

**The Identification of Media Viewed Using Windows Media Player within  
Windows 7 Through the use of Cluster Analysis**

**University of Northumbria**  
**School of Computing, Engineering and Information Sciences**

I declare the following:

(1) The material contained in this dissertation is the end result of my own work, and that due acknowledgement has been given in the bibliography and references to **ALL** sources; be they printed, electronic or personal.

(2) The Word Count of this Dissertation is:

(3) That unless this dissertation has been confirmed as confidential, I agree to an entire electronic copy or sections of the dissertation to being placed on the eLearning Portal (Blackboard), if deemed appropriate, to allow future students the opportunity to see examples of past dissertations. I understand that if displayed on eLearning Portal it would be made available for no longer than five years, and that students would be able to print off copies or download.

(4) I agree to my dissertation being submitted to a plagiarism detection service, where it will be stored in a database and compared against work submitted from this or any other School or from other institutions using the service. In the event of the service detecting a high degree of similarity between content within the service this will be reported back to my supervisor and second marker, who may decide to undertake further investigation that may ultimately lead to disciplinary actions, should instances of plagiarism be detected.

(5) I have read the Northumbria University/CEIS Policy Statement on Ethics in Research and Consultancy and I confirm that ethical issues have been considered, evaluated and appropriately addressed in this research.

Signed:

Date: 08/04/2011

**Acknowledgements**

I would like to thank my project supervisor Dr. Christopher Laing for the guidance, support and general mocking which has helped me throughout this project. I would also like to thank Philip Anderson for his general guidance throughout this project.

Many thanks to Abby Popplestone for help with proof reading. Thanks to Special Detective Allan Hay from the Northumbria High Tech Crime unit on his input into this project.

A special thanks to friends and family for putting up with me ignoring them while I completed this project.

**Abstract**

Windows Media Player (WMP) version 12.0 is the latest version available solely within the Windows® 7 Operating System. The latest version of WMP is capable of viewing more file types than any other version, an example of this is that WMP version 12.0 is capable of viewing .avi files by default whereas version 11.0 is not. An investigation found that WMP records information about media viewed within the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf* files. The investigation also indicated that currently little to no research is available about how WMP records entries within both files, as well as what appears to be a no current solution. With no current known solution to the extraction of information about media viewed using WMP within Windows® 7, a software application is deemed necessary. The development of software will enable the prioritising of cases in an effort to reduce the national backlog of computer forensic cases. Through research into the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf* files, patterns were discovered prior, throughout and after media entries that exist within both files. The research conducted enabled a design to be constructed to aid in the implementation of the software. A proof of concept was developed to extract media entries from both files and create a report. The scope of the project defined the testing strategy to be used. Testing appears to show that although a proof of concept the software is capable of fulfilling its requirements in both build quality and fitness for purpose.

**Keywords:** Windows Media Player, WMP Extractor, Prefetch, Media, Computer Forensics, Cluster Analysis

## Contents

1. Introduction.....	9
1.1.Discussion of the background to the project.....	9
1.2.Purpose of the Project .....	10
1.3.Project Scope .....	10
1.4.Project Limitations.....	11
1.5.Subject to be considered.....	11
1.6.Nature of the Study and Application Area .....	11
1.7.Principle Conclusions and Recommendations.....	12
1.8.Structure of the Report.....	12
2. Analysis .....	14
2.1.Background to Windows Media Player .....	15
2.1.1.Windows® 7 N .....	15
2.2.The Significance of Windows Media Player in a Forensics Investigation .....	15
2.3.Initial Research .....	16
2.3.1.Windows Media Player Database .....	16
2.3.2.Windows Media Player Prefetch.....	17
2.4.Background to Windows Media Player in Windows 7 .....	18
2.5.Overview of Windows Media Player .....	19
2.6.Windows 7 N.....	19
2.7.Overview of CurrentDatabase_372.wmdb.....	21
2.7.1.Audio Layout .....	23
2.7.2.Image Layout .....	26
2.7.3.Video Layout .....	29
2.7.4.Deleting Media from the Windows Media Player Library .....	31
2.8. Overview of WMPLAYER.EXE-xxxxxxxxx.pf.....	31
2.8.1.Background to Windows Prefetch files.....	31
2.8.2.Windows Media Player Prefetch in Windows 7 .....	32
2.9.Current solutions .....	34
2.9.1.Windows Media Player Database .....	34
2.9.2.Windows Prefetch .....	35
2.10.Possible Approaches .....	35
2.10.1.Cluster Analysis .....	37

2.11. Software Requirements and Justification .....	37
2.11.1. Extract information about media viewed upon Windows 7 .....	38
2.11.2. Ensure the evidential integrity is maintained when extracting information .	38
2.11.3. Produce a report for easy analysis .....	40
2.11.4. Have Graphical User Interface (GUI) to allow for ease of use .....	40
2.12. Identification of Tools and Techniques .....	41
2.13. Analysis Overview .....	42
3. Synthesis .....	44
3.1. The problem .....	45
3.2. Design Overview .....	46
3.3. Unified Modelling Language (UML) .....	47
3.3.1. What is UML? .....	47
3.3.2. Reason for using UML .....	47
3.3.3. Chosen UML diagrams .....	48
3.3.4. Architecture of WMP Extractor .....	49
3.3.5. Use Case Diagram .....	50
3.3.6. Use Case Description .....	53
3.3.7. Pseudo Code .....	53
3.3.8. Class Diagrams .....	54
3.4. Design of WMP Extractor .....	55
3.4.1. Windows Media Player Database .....	55
3.4.2. Windows Prefetch .....	57
3.5. Human Computer Interface (HCI) .....	58
3.6. Design of the Graphical User Interface (GUI) .....	58
3.7. Design Implementation .....	61
3.7.1. PrefetchSearch – class .....	63
3.8. Testing .....	64
3.8.1. Testing Strategy .....	64
3.8.2. Windows Media Player Database Testing .....	65
3.8.3. Windows Media Player Prefetch Testing .....	67
3.8.4. Test Cases .....	68
3.8.5. User Testing .....	69
4. Evaluation .....	71
4.1. Evaluation of Fitness for Purpose .....	72

4.2. Evaluation of Build Quality .....	78
4.3. Evaluation of Project Process.....	79
4.3.1.Evaluation of Tools and Techniques .....	82
4.3.2.Evaluation of Testing .....	84
4.4.Further Work .....	85
4.4.1.Alternative uses forWMP Extractor.....	86
4.4.2.Further Work into WMP Extractor.....	87
4.5.Conclusions.....	89
4.6.Recommendations.....	91
5. Appendix.....	94
5.1. Terms of Reference.....	95
5.2. Ethics Form .....	104
6. References.....	106
7. Bibliography.....	112
7.1.Images .....	113
7.2.Use Case Descriptions.....	117
7.3.Pseudo Code.....	121
7.3.1.WMPExtractor .....	121
7.3.2.GraphicalUserInterface .....	121
7.3.3.WMDNFileSignatures .....	122
7.3.4.WMDBExtractor.....	122
7.3.5.PrefetchExtensions.....	122
7.3.6.PrefetchSearch .....	123
7.3.7.HexConverter .....	123
7.3.8.DateAndTime.....	123
7.3.9.FileSize .....	124
7.3.10.OutputToFile .....	124
7.3.11.HashValue .....	124
7.4.Class Diagram.....	125
7.5.Design Implementation.....	126
7.5.1.WMPExtractor– class .....	126
7.5.2.GraphicalUserInterface – class .....	126
7.5.3.WMDBFileSignatures– class.....	128
7.5.4.WMDBExtractor– class .....	128
7.5.5.PrefetchExtensions – class.....	129

7.5.6.HexConverter – class .....	129
7.5.7.DataAndTime – class .....	130
7.5.8.FileSize - class .....	131
7.5.9.OutputToFile – class .....	131
7.5.10.HashValue – class .....	132
7.6.Testing .....	134
7.6.1.Windows Media Player Database Testing.....	134
7.6.2.Windows Media Player Prefetch Testing.....	136
7.6.3.Test Cases .....	138



## 1. Introduction

### 1.1. Discussion of the background to the project

Windows® 7 includes the new version of Windows Media Player (WMP), this version of WMP (version 12.0) is only available for Windows® 7. The new version of WMP is capable of viewing more different media types than any other version of WMP. WMP within Windows® 7 has the default ability to view .avi files, compared to WMP version 11.0 which required the installation of codecs to support .avi files. Storage capacity in the form of hard drives and external media has increased significantly over the past twenty years (Sandler, C 2007). The increase in hard drive capacity has enabled the storage of an increased amount of media upon computers both internally and externally. Allan Hay from the Northumbria High Tech Crime Unit suggested that, 90% of the computer forensic cases that police see involve the possession of illegal media upon the suspect's computer as well as external storage such as USB hard drives (Hay, A.S 2011). Windows® 7 currently comes installed on almost all new laptops and desktops available from major retailers such as DELL and HP. These factors combined will inevitably lead to the police forensically examining an increasing number of computers with Windows® 7 as their primary Operating System. WMP within Windows® 7 stores information about media viewed, information is reordered within two main locations; the WMP database file *CurrentDatabase\_372.wmdb* and the WMP Prefetch files *WMPLAYER.EXE-xxxxxxx.pf*. Information recorded within both of these files includes:

1. File name
2. Full file directory
3. Information from the "Details" section of the file properties
4. Size of file in bytes (not 'Size of disk')
5. Size of file in pixels
6. Date added to WMP
7. Date last viewed
8. Number of times viewed

Through research it has been discovered that there are currently limited solutions to the recovery of information about media viewed using WMP; across different versions of the Windows® Operating System, this has been noticed by the forensic community in such places as computer forensic forums (Forensic Focus, a 2006). While there are some solutions to the problem of extracting information about media viewed using WMP within certain versions of Windows®, after research it appears

that there is currently no solution to the extraction of information from within Windows® 7.

### 1.2.Purpose of the Project

Information about media viewed using WMP is fragmented throughout Windows® 7, both the WMP database and WMP Prefetch files contain information such as that noted above about media viewed using WMP. The aim of the project is to extract information from both the WMP database and WMP Prefetch files, bring the information together, and produce a report. Information contained within the report will enable the forensic examiner to build up a picture of what media the suspect has been viewing. By enabling the examiner to see what media the suspect has been viewing, it will allow the prioritising of cases depending on the results returned. If several computers have been seized with regards to the suspected viewing of illegal media, the analysis of both the WMP database and WMP Prefetch may enable the forensic examiner to determine which computers have been used and which have not to view illegal media. The results produced from the analysis would enable the examiner to then prioritise their case load. The prioritising of caseloads would help reduce the “ten month national backlog” (Kennedy, 2009). A solution for the extraction of media viewed using WMP upon a suspect’s machine would allow an examiner to potentially determine if a suspect has been involved in the viewing of illegal media without the need to seize the computer. If an analysis of the WMP database and WMP Prefetch files shows that a suspects has *not* been viewing illegal media using WMP, it may prevent the need for the computer to be seized which has the potentially to save tens of hours of examination time had the machine been seized.

### 1.3.Project Scope

In order to provide the computer forensic community with a solution to the problem mentioned above, areas of research need to be conducted, a solution devised and then tested to ensure it functions as expected for this to be accomplished. Through research conducted there does not appear to be any information by other researchers, or released by Microsoft® on how information is stored within the WMP database. Research will need to be conducted into identifying the patterns on how WMP stores information about media viewed within the *CurrentDatabase\_372.wmdb* database as well as within the *WMPLAYER.EXE-xxxxxxx.pf* Prefetch files. Once the patterns have been identified a solution will need to be formulated. The solution will need to ensure that all information identified as relevant from within the WMP database and WMP Prefetch files is extracted accurately to ensure evidential is maintained. After a solution has been formulated and implemented it will need to be tested to ensure that it performs as expected with results produced as well as its usability.

The solution to the extraction of information about media viewed using WMP within Windows® 7, will need to extract information from both the WMP database and WMP Prefetch files and output it to a report. During the extraction of the information and outputting of it to the report evidential integrity must be maintained to ensure validity of results. Whilst a proof of concept, the solution will need to provide the examiner will a Graphical User Interface (GUI), this will enable to for less training to be required to use the solution as well as enabling for more control during the

analysis. The combination of these requirements will provide the computer forensic community with the ability to analyse media viewed using WMP.

#### 1.4. Project Limitations

As with the construction of any software a suitable programming language must be used, the programming language must be able to handle the requirements of the software i.e. run on a specific platform such as Windows®. The Java programming language was used to construct the software. Java was chosen as it was deemed the most suitable language for this project. While Java was a suitable language within the scope of this project it is not without limitations. An example of one the limitations is that Java requires the Java Virtual Machine (JVM) to be installed to run. The JVM does not come installed with Windows® (unless installed by a third part vender). Without the JVM the software application that has been designed, created and tested throw-out this project is unusable. While this is a limitation; it was deemed to be an acceptable one as if the suspect's machine does not have the JVM installed the examiner will not be able to perform a live analysis but, will be able to perform a static analysis upon a computer than has the JVM installed.

Another limitation of this project is that the software constructed for this project is currently only able to extract information about media viewed using WMP from within Windows® 7 using WMP 12.0. As mentioned above through the research conducted by the author there was no current solution found for WMP version 12.0 which is only available within Windows® 7. While this is a limitation; it was by design. Future work within the area of information extraction about media viewed using WMP within the Windows® Operating System could overcome this limitation.

#### 1.5. Subject to be considered

As mentioned above the author conducted research into the current solutions to the extraction of media viewed using WMP within Windows®. Research concluded that while solutions do exists for previous versions of Windows® such as XP, no known current solution exists for WMP within Windows® 7. With Windows® 7 being the latest version of Windows® from Microsoft®, only it is capable of running WMP 12. Due to what appears through research to be a gap within the computer forensic community's ability to extract information about media viewed using WMP within Windows® 7, this project will concentrate solely in this subject area.

#### 1.6. Nature of the Study and Application Area

Through research conducted by the author, it has not been possible to find any current research into the extraction of information about media viewed using WMP within Windows® 7. There is also currently no information available Microsoft® detailing how information is stored within either the WMP database or WMP Prefetch files. Due to the lack of information available about how information is stored within the WMP database and WMP Prefetch files, from both research conducted and information available from Microsoft®, this project will concentrate on understanding how this is done in order to provide a solution. Research will be conducted into what information is available within the WMP database and that available within the WMP Prefetch files. The research will go on to provide information on patterns within each

of the files that exist prior, throughout and after media entries that exists within the WMP database and WMP Prefetch files. Using these known patterns will enable the development of a software application that will aid in the extraction of the information contained within the WMP database and WMP Prefetch.

### 1.7.Principle Conclusions and Recommendations

Prior to the undertaking of this project research as mentioned above, was conducted to ascertain the need for a solution to the extraction of information about media viewed using WMP within Windows®7. It was concluded through research that no current solution was available and that a solution would be advantages to the computer forensic community. Through the continued research patterns were discovered that exist prior, throughout and after each media entry within the WMP database and WMP Prefetch files. Once it had been ascertained how information is stored within the WMP database and WMP Prefetch files a design was constructed and software developed. The software developed within the scope of this project has shown that that it meets all the requirements set out within the *Software Requirements and Justification* (2.11). Tests conducted throughout aimed to provide a balanced testing of the software within its fitness for purpose and build quality. While the software lived up to both the desired fitness for purpose and build quality, tests used were designed for use within the scope of the project. Should this project be expanded upon, more relevant testing would need to be undertaken. Due to this project being what appears to be unique research, it was not without limitations. Limitations with the project can be addressed by recommendations such as future work. As discussed above one of the limitations of this project was that the software created requires the JVM. While this could be overcome by performing an analysis of the WMP database and WMP Prefetch file upon an examiners machine, it none the less reduces the overall functionality of the software. A recommendation to solve this limitation is to use another programming language that does not require the JVM such as C#; this would ensure that an examiner is able to perform an analysis without the need for support from third party software. Another limitation within this project is that the software is only capable of performing an analysis on the WMP version 12.0 database; if an examiner investigates a case that involves any other versions of WMP they will be unable to perform an analysis. A recommendation to solve this issue would be the further continued research into different versions of the WMP database, and implementing the research into the current software application.

### 1.8.Structure of the Report

As discussed above this report aims to provide a solution to the extraction of media viewed using WMP within Windows® 7. This report will focus on the identification of information about media viewed using WMP within Windows® 7, and the development of the application for the extraction of the information. The report will cover three main areas including analysis, synthesis and evaluations and conclusions. Each section will be broken down; each concentrating on specific tasks that are required to complete the project. The analysis will consist of the identification of information about media viewed using WMP. It will discuss what specific information is available from within the WMP database and WMP Prefetch files, the identification of patterns found with how information is stored within each file, and continue on to

discuss what of the information is relevant in the scope of this project and that; that will be extracted. The analysis will then discuss, current known solutions to the extraction of information about media viewed using WMP across the Windows® platform. The discussion of known current solutions will discuss the extraction of information from the WMP database and the WMP Prefetch files and their weakness with regards to media information extraction. Possible approaches will be considered for the extraction of information about media viewed using WMP and will take into account the patterns of information discussed earlier within the analysis. The software requirements and justification section will discuss the requirements that the software application must have, and the appropriateness of these requirements. To construct the software application the analysis will discuss the tools and techniques to be used such as the programming language.

The synthesis will contain the design of the application, including the chosen design language and the design diagrams to be used in the construction. Once a suitable design has been constructed the synthesis will discuss the implementation. The implementation will discuss how the software was constructed, any issues that were encountered during the software implementation and how these were resolved. Once the software has been constructed the synthesis will discuss a test plan that will determine if the results returned from the extraction of information from the WMP database and WMP Prefetch are accurate and that the software performs as expected.

The final section of this project will discuss the evaluation of project. The evaluation will discuss the software application fitness for purpose and its build quality. To learn from this project to improve on future work the project process will be evaluated, the tools and techniques used throughout the project will be evaluated to see if a more effective approaches could be undertaken if the process were to be repeated or future work be undertaken. With this project being a proof of concept the synthesis will discuss further work that could be carried out in order to improve the functionality of the software and increase its usability as a computer forensic application. As well as the further work that could be carried out from this project such as improvements to the application, the synthesis will discuss alternative uses for the software created during the implementation. Conclusions and recommendations will then be drawn to sum up the overall project.

## **2. Analysis**

## 2.1. Background to Windows Media Player

Windows Media Player (WMP) is an application developed by Microsoft Corporation to view media including audio, video, DVD's, internet radio, and images upon the Microsoft® Windows® Operating System (Minasi, M. Mueller, J.P 2007). WMP is also available on Pocket PC, Windows Mobile and MAC OS which development has been discontinued but is still available for download from the Microsoft® website (Microsoft® Corporation, e 2009). WMP was first introduced in Microsoft® Windows® 3.00a in 1991 (Microsoft® Corporation, b 2010), since its release there have been several versions (Microsoft® Corporation, e 2009). Some versions of WMP are exclusive to each version of Microsoft® Windows® (Microsoft® Corporation, e 2009). The latest version of WMP (version 12) was released alongside Windows® 7 and is currently not available for any other version of Windows® (Microsoft® Corporation, e 2009). Whilst the primary function of WMP is to view media, it has several other functions including: ripping<sup>1</sup> music from CD's and burning<sup>2</sup> music to CD's introduced in version 7 (Microsoft® Corporation, f 2010); as well as synchronizing media with selected media players and the purchase of media online.

### 2.1.1. Windows® 7 N

Windows® 7 is available without WMP preinstalled in a version called *Windows® 7 N*, this was created to comply with the European Commissioners ruling that Microsoft® was abusing its monopoly in the market by supplying Windows® with built in features such as WMP (CNN, 2004). While this version is available many suppliers of Microsoft® Windows® will not stock Windows® 7 N as there is “no demand” for it (CNET, 2005). While Windows® 7 N does not come with WMP it is still possible to download and install the latest version from the Microsoft® website (Microsoft® Corporation, d 2010). Windows® 7 N is not currently stocked in store or online at major retailers. While this may change, it shows that the version of Windows® 7 that includes WMP is currently the dominant, and that there is a need for a software tool to extract information from the WMP database.

## 2.2. The Significance of Windows Media Player in a Forensics Investigation

Microsoft® Windows® is currently the dominant desktop operating system with an estimated 90%+ share throughout their operating systems range within the home and workplace combined (Market Share, 2010). This coupled with WMP coming preinstalled on all versions of Windows® 7 (excluding Windows® 7 N), and being the default media player may increase the chance of WMP being used to view media be it legal or illegal.

Windows® 7 is the latest Operating System (OS) from Microsoft®, WMP comes preinstalled on all versions of Windows® 7 (except Windows 7 N). The new version of WMP is able to view more types of media than ever before “straight out of the box”. With more media types being supported it is less likely that a user will need to

---

<sup>1</sup> Ripping music from a CD is the act of copying onto the hard drive of the computer.

<sup>2</sup> Burning music is the act of storing music on to a physical media disc such as a CD.



install a different media player to support their media type such as VLC media player; which has support for a large array of media types (Video Lan, 2010). With the possibility of WMP being used to view media more often than not the WMP database will provide a rich source of information for a forensic investigation.

With record sales of Windows 7 (The Telegraph, 2009), Microsoft's® large market share and WMP installed on all versions of Windows® 7 (excluding Windows® 7 N), it could be argued that WMP is possibly the most used media player. There is currently limited solutions to extracting information about what media has been viewed using WMP from within Microsoft® Windows® 7.

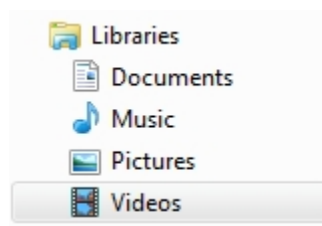
### 2.3.Initial Research

This section will discuss the initial research that has shown that WMP within Windows® 7 records a significant amount of Metadata<sup>3</sup> about media viewed using WMP. There are two main areas that hold significant information for use in an investigation, not only when viewed alone but when combined to create a better idea of what media the user has been viewing.

#### 2.3.1.Windows Media Player Database

Initial investigation has revealed that when a user imports media into WMP it is recorded within the WMP database file, the database file is static meaning that entries are only added to the database never removed. To import media in to WMP in Windows® 7 a user can save a file type associated with WMP to their libraries<sup>4</sup>. This will automatically get added to WMP if WMP is running or once WMP is started after adding the media to the library. Another way a user can import media into WMP is drag and drop media on to WMP.

*Figure 1* below shows the default libraries that are pre-setup all on versions of Windows® 7.



*(Figure 1 – shows the default libraries in Windows® 7)*

Entries are never removed from the WMP database file it will provide a history of what media the user has been viewing. It must be noted that if a user just plays a song using WMP by double clicking it will *not* be added to the database file. The WMP database is where the most information is available to the forensic examiner about the activities of the user. The WMP can store information about:

#### 1. File name

---

<sup>3</sup> Metadata is data about data, e.g. the date stamp of a Word document would be considered Metadata.

<sup>4</sup> Libraries are the default folders where the user stores their files these include, Documents, Music, Pictures and Videos, a user is able to create new libraries as well as delete default ones.



2. Full file directory
3. Information from the “Details” section of the file properties
4. Size of file in bytes
5. Size of file in pixels
6. Date added to WMP
7. Date last viewed
8. Number of times viewed

With such large amounts of data recorded about the user’s activities it allows the forensics examiner to build up a picture of what media the user has been viewing. With all this information it may help potentially proves or disproves if a user has been viewing illegal media.

### 2.3.2.Windows Media Player Prefetch

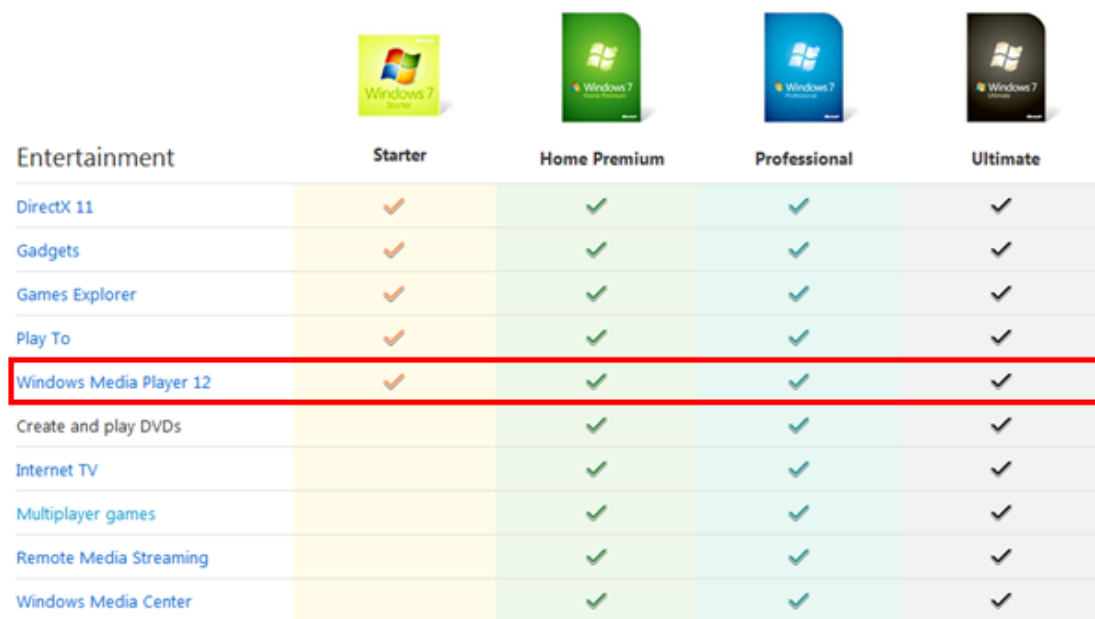
The WMP database is not the only place that Windows® 7 records information relating to media viewed using WMP. Other places where information relating to media viewed using WMP include: Windows® Prefetch as shown within *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8.1) files which after initial investigation records files viewed using WMP. The Windows Media Player Prefetch file has a major advantage over the WMP database file in that it records files viewed using WMP that are not necessary recorded within the WMP database. A user has several ways they can view media using WMP without importing the file into WMP. A user is able to double click the media file, which if that media file type is associated with WMP, WMP will open and play the media file. It is also possible for the user to right click the media and select “Play” which will result in the same actions as if the user double clicked the media file. The other option is for the user to right click the media of which it is possible to select more than one file and select “Add to Windows Media playlist”, however this will only be available if the media is associated with WMP.





With the lack of current tools in place to extract information about media viewed using WMP from Windows® 7, a forensics examiner will have to manually parse the database file through a Hex editor to view the information for it to be extracted. With the development of tool to extract relevant information from within the WMP database it will allow the forensic examiner to potentially speed up the examination time or allow non-forensic personal to complete the task.

## 2.4. Background to Windows Media Player in Windows 7

Each version of Windows® 7 (excluding Windows® 7 N) comes preinstalled with a copy of Windows Media Player (WMP) (Current version 12.0.7600.16667). Each user account has its own WMP database named *CurrentDatabase\_372.wmdb* in the directory *root:\Users\%USERS%\AppData\Local\Microsoft\Media Player\*, this file contains information including that shown in *Overview of WMPLAYER.EXE-xxxxxxx.pf(2.8)* about media imported into WMP.

There are four versions of Windows® 7 available for purchase/that come preinstalled on new computers throughout Europe. *Figure 2* below shows the differences between each version; it also shows that each version of Windows® 7 comes with WMP.



	 Starter	 Home Premium	 Professional	 Ultimate
Entertainment				
DirectX 11	✓	✓	✓	✓
Gadgets	✓	✓	✓	✓
Games Explorer	✓	✓	✓	✓
Play To	✓	✓	✓	✓
Windows Media Player 12	✓	✓	✓	✓
Create and play DVDs		✓	✓	✓
Internet TV		✓	✓	✓
Multiplayer games		✓	✓	✓
Remote Media Streaming		✓	✓	✓
Windows Media Center		✓	✓	✓

(Figure 2 –shows the WMP version 12.0 come with all versions of Windows® 7)  
(Microsoft® Corporation, c 2009)

- Windows 7 Home Starter 32bit: Windows 7 Starter is designed for netbooks; it is the version with the least amount of features.
- Windows 7 Home Premium 32/64bit: Windows 7 Premium is currently the most likely to come preinstalled on a new laptop.
- Windows 7 Professional 32/64bit: Windows 7 Professional is designed towards businesses with features such as Group Policy controls, Encrypting File System (EFS) and Offline folders.
- Windows 7 Ultimate 32/64bit: Windows 7 Ultimate is the only version with all features which include Window BitLocker file encryption.

<sup>5</sup> %Users% denotes the users profile name e.g. *root\Users\James\AppData\Local\Microsoft\Media Player*

To show that each version including both the 32 and 64 bit versions of Windows® 7 record entries within the WMP database in the same way, all versions will be tested using the same criteria. These tests will show if the databases are created the same to rule out the need to design the software to cope with several versions.

To investigate the WMP database file several copies of the file must be examined in its various states of use by a test user. The hard drive that Windows 7 will be installed upon must first be forensically wiped using Darik's Boot and Nuke (DBAN) (Darik's Boot and Nuke, 2010). This ensures that there will be no data left upon the hard drive that could possibly interfere with the results. Forensically wiping a hard drives is considered standard practice as any changes to data could render it inadmissible in court.

## 2.5. Overview of Windows Media Player

The first time WMP is run it requires the user to make a selection about the type of settings that will be in place, for this investigation the "recommended settings" will be used. The use of the "recommended settings" will not have an effect upon how the data is recorded within the WMP database. Once Windows® 7 has been installed on a computer, a copy of the WMP database file is placed within the directory *root:\Users\%USERS%\AppData\Local\Microsoft\Media Player*, the file is located there even if WMP has *not* been run. The WMP database that is located within *root:\Users\%USERS%\AppData\Local\Microsoft\Media Player* before WMP is run for the first time has a MD5 hash value of "180FF61FA14EFD4D09DDA14040DAA41E" on all versions of Windows® 7. With the hash value for WMP database the same prior to WMP being run, it shows that the file is always created the same proving that any entries within the file were created by the user as all databases were empty.

*Figure 3* below shows the differences in the WMP database after installation of all different versions of Windows® 7. The differences were calculated by comparing the hash values of all versions of which none were found.

Windows 7 Version	File name	Differences
Windows 7 Starter 32bit	CurrentDatabase_372.wmdb	None
Windows 7 Home Premium 32/64bit	CurrentDatabase_372.wmdb	None
Windows 7 Professional 32/64bit	CurrentDatabase_372.wmdb	None
Windows 7 Ultimate 32/64bit	CurrentDatabase_372.wmdb	None

*(Figure 3 –shows the differences between the WMP databases from different versions of Windows® 7)*

## 2.6. Windows 7 N

As mentioned in *Background to Windows Media Player* (2.1) Windows® 7 N does not come preinstalled with WMP but the user is able to download it from the Microsoft® website (Microsoft Corporation, e 2011). Due to WMP being installed after Windows® 7, the WMP database file will have a different Hash value because of the different created date. This report will concentrate on the versions of Windows® 7 that come preinstalled with WMP, these versions of Windows® 7 are more likely to

be encountered by forensic examiners due to Windows® 7 N having a smaller market share(CNET, 2005).

### 2.7.Overview of *CurrentDatabase\_372.wmdb*

Below will discuss patterns of how information for audio, image and video entries are stored within the WMP database. When an entry is written to the WMP database file, it does not permanently stay at the same file offset. Entries within the database file move around depending on several factors.

1. When any media file is viewed
2. When new entries are added to the database
3. When files are viewed

There may be other factors that cause entries within the WMP database to move around but these are unknown as there is no documentation available from Microsoft®. Although the above factors do cause entries to be moved around the WMP database, it is not always the case. As mentioned it is not known how and when WMP decides to move entries around due to the lack of documentation on the WMP database available from Microsoft®.

An example of entries moving within the WMP database is shown below, *Figure 4* shows the location of an image entry at file offset 978928.

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	10	0123456789ABCDEF0
0x0EEFF0	0000	0000	0000	0000	0000	0000	9000	0000	36	..... 6
0x0EF001	F8EE	77DA	0440	002C	0000	0000	0000	00B2	04	øiwŮ. @.,.....².
0x0EF012	0000	01D0	0F00	0000	0000	2500	0000	08B9	00	...Đ.....§.....¹.
0x0EF023	CAE6	0401	0000	0087	A438	04B5	9E00	0075	E8	Êæ.....+x8.µž...uè
0x0EF034	0C00	0000	0000	0003	0000	0004	0000	0000	00	.....
0x0EF045	0000	0000	0000	0000	004A	E7B2	6FCB	B8BB	40	.....Jç°oĚ.»@
0x0EF056	93F3	FAC5	F00F	A203	0000	0000	0000	0000	00	"óúĀđ.¢.....
0x0EF067	0000	0000	0000	0005	93AB	0749	2A4C	4F96	B4	....."«.I*LO-'
0x0EF078	8F73	16A7	2C2A	2F86	0003	639A	0000	D807	00	s.Š,*/+..cš...Ø..
0x0EF089	0003	0000	000E	0000	0050	EE46	7A24	EBCB	01	.....PîFzšĚĚ.
0x0EF09A	50EE	467A	24EB	CB01	3200	0000	0000	0000	00	PîFzšĚĚ.2.....
0x0EF0AB	0000	0000	8007	0007	801E	8025	0025	8029	80	....€...€.€§.§€)€
0x0EF0BC	2D00	2D80	4D00	4D00	4D00	4D00	4465	7365	72	.-.€M.M.M.M.Deser
0x0EF0CD	7400	443A	5C50	6963	7475	7265	735C	4465	73	t.D:\Pictures\Des
0x0EF0DE	6572	742E	6A70	6700	436F	7262	6973	006A	70	ert.jpg.Corbis.jp
0x0EF0EF	6700	6A70	6700	A920	436F	7262	6973	2E20	20	g.jpg.© Corbis.
0x0EF100	416C	6C20	5269	6768	7473	2052	6573	6572	76	All Rights Reserv
0x0EF111	6564	2E00	0000	0000	0000	0000	0000	0000	00	ed.....
0x0EF122	0000	0000	0000	0000	0000	0000	0000	0000	00	.....

(*Figure 4* – shows the location of a .jpg media entry)

After creating an audio entry within the WMP database by adding an audio file to the WMP library, the image entry has now been moved to file offset 1007607. *Figure 5* below shows the same file offset 978928 where the original image entry was located

prior to adding the audio entry. The information shown in *Figure 5* below where the original image entry was located prior to the adding of the audio entry is unknown, due to no documentation from Microsoft® it has not been possible to distinguish what this represents.

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	10	0123456789ABCDEF0
0x0EEFF0	0000	0000	0000	0000	0000	0000	9000	0000	75	.....u
0x0EF001	4C14	1526	0460	00F5	0200	0000	0000	0000	00	L...ä.`.õ.....
0x0EF012	0000	0000	0000	0100	0000	0000	0000	00E0	4D	.....àM
0x0EF023	0045	8000	9EB5	043E	C2F0	4500	0000	0100	00	.E€.žµ.>Ã&E.....
0x0EF034	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF045	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF056	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF067	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF078	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF089	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF09A	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF0AB	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF0BC	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF0CD	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF0DE	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF0EF	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF100	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF111	0000	0000	0000	0000	0000	0000	0000	0000	00	.....
0x0EF122	0000	0000	0000	0000	0000	0000	0000	0000	00	.....

(Figure 5 – shows the location where a .jpg media entry was prior to being moved)

For each of the audio, image and video entries within the WMP database four instances of the file name occur. In each of the audio, image and video entries the first instance contains the most information relating to the users media viewing. Below discusses the audio, image and video entries in turn, each section describes where information is located after file signature entries. File signature entries are clusters of Hex values that are located prior to information about a file entry. After a prefixed number of bytes within the WMP database after the file signature entry specific information is located. An example is that the first audio entry file signature starts with F272 5305 55. Thirty nine bytes after the file signature entry is the size of the file in bytes, the file size takes up three bytes within the WMP database. As discussed above there are four instances that occur for each media type, through research into the WMP database it has been noticed that these entries can become repeated, it is not clear as to why this happens. To try to get an understanding as to why this happens the author tried to replicate this ‘anomaly’ but was unable to find a root cause. While the author does not foresee this causing issue with the project, it is worth noting in case an examiner performs an analysis and multiple results are returned.

### 2.7.1.Audio Layout

The first entry within the WMP database contains the most information about the file viewed. *Figure 6* below shows a visual representation of an audio entry.

1. File entry signature
  - a. First audio entries start with the Hex values F272 5305 55
2. File size in bytes (39 bytes after file entry signature)
  - a. Actual file size *not* size of disk
  - b. File size takes up three bytes
3. Date and time media file added to WMP (115 bytes after file entry signature)
  - a. Date and time media file added takes up eight bytes
4. Date and time media file last viewed (123 bytes after file entry signature)
  - a. Date and time media last viewed takes up eight bytes
5. Number of times media file viewed (205 bytes after file entry signature)
  - a. Number of times media file viewed takes up one byte
  - b. For this to be incremented the “Seek” bar must reach the end
  - c. Skipping to almost the end using the “Seek” bar counts as a viewed
6. File name (285 bytes after file entry signature)
7. Full media file name and path
8. Extra details about media file

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF
0x106000	8EE7	4B22	E004	4000	1E00	0000	0000	0000	žçK"a. @.....
0x106010	0504	0000	01D0	2500	0000	0000	4300	0000	.....Đš.....C..
0x106020	C008	0077	0540	0006	F800	0100	0000	C7A9	À..w.@..ø.....Çø
0x106030	7202	739E	0000	0100	0000	F272	5305	5500	r.sž.....šrS.U
0x106040	0000	0000	1000	8000	00AA	0038	9B71	0000	.....ē..8>q..
0x106050	0000	0000	0000	60EA	0000	1B9A	0000	D807	.....ē..š..ø.
0x106060	0000	974F	0500	F164	8C00	0000	0000	00EE	..-O..řde.....i
0x106070	0200	BC7D	60D1	23E3	E24B	86A1	48A4	2A28	..¼) `N#āāK+;H* (
0x106080	441E	0000	0000	0000	0000	0000	0000	0000	D.....
0x106090	0000	F3A0	0F4F	953D	1A47	B0D2	9DCB	30A9	..ó .O*=.G°ò Èøø
0x1060A0	BBAE	77FF	8439	C0FB	3545	8A23	3B50	0620	»øwŷ„9Āū5EŠ#;P.
0x1060B0	645C	9000	7645	02B7	CB01	9000	7645	02B7	d\ ..vE..ē. ..vE..
0x1060C0	CB01	0000	0000	0000	0000	0000	0000	0000	ē.....
0x1060D0	0000	0000	0000	0000	0000	0100	0000	0100	.....
0x1060E0	0000	0100	0000	355D	A05F	82A6	F64A	96F7	.....5] _ , !ōJ-+
0x1060F0	0773	E42D	4D16	0000	0000	0000	0000	0000	..sä-M.....
0x106100	0000	3200	0000	0000	0000	0000	0000	0000	..2.....
0x106110	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x106120	0000	0000	0000	C7A9	7202	739E	0000	0000	.....Çør.sž....
0x106130	0000	0080	0800	0880	2980	2D80	3100	3100	...ē...ē)ē-ē1.1.
0x106140	3180	4B80	5680	6180	6C80	7780	8F80	9A00	1eKeVeaeleWe eš.
0x106150	9A00	9A80	B280	D480	F380	1581	4B61	6C69	š.šē*ēōēōē. Kali
0x106160	6D62	6100	433A	5C55	7365	7273	5C4A	616D	mba.C:\Users\Jam
0x106170	6573	5C4D	7573	6963	5C4B	616C	696D	6261	es\Music\Kalimba
0x106180	2E6D	7033	006D	7033	006D	7033	0041	2E20	.mp3.mp3.mp3.A.
0x106190	4361	7274	6879	2061	6E64	2041	2E20	4B69	Carthy and A. Ki
0x1061A0	6E67	736C	6F77	0045	6C65	6374	726F	6E69	ngslow.Electroni
0x1061B0	6300	4D72	2E20	5363	7275	6666	004E	696E	c.Mr. Scruff.Nin
0x1061C0	6A61	2054	756E	6100	4D72	2E20	5363	7275	ja Tuna.Mr. Scr
0x1061D0	6666	004E	696E	6A61	2054	756E	612A	3B2A	ff.Ninja Tuna*;*
0x1061E0	4D72	2E20	5363	7275	6666	004D	722E	2053	Mr. Scruff.Mr. S
0x1061F0	6372	7566	6600	4D72	2E20	5363	7275	6666	cruff.Mr. Scruff
0x106200	4E69	6E6A	6120	5475	6E61	3030	3100	456C	Ninja Tuna001.El
0x106210	6563	7472	6F6E	6963	4D72	2E20	5363	7275	ectronicMr. Scr
0x106220	6666	4E69	6E6A	6120	5475	6E61	3030	3100	ffNinja Tuna001.
0x106230	4B61	6C69	6D62	614D	722E	2053	6372	7566	KalimbaMr. Scruf
0x106240	664E	696E	6A61	2054	756E	6130	3031	004D	fNinja Tuna001.M
0x106250	722E	2053	6372	7566	664D	722E	2053	6372	r. ScruffMr. Scr
0x106260	7566	664E	696E	6A61	2054	756E	6130	3031	uffNinja Tuna001
0x106270	004D	5033	0000	0000	0000	0000	0000	0000	.MP3.....

(Figure 6 – shows the layout of information of contained within the WMP database for audio entries)

The second, third and fourth entry within the WMP database are grouped within close proximity of each other. The number of bytes between the first entry of an audio entry and the second, third and fourth varies, there does not appear to be a noticeable pattern. Figure 7 below shows a visual representation of other data available within the WMP database for an audio entry.

1. File entry signature
  - a. Second audio entries start with the Hex values C000 0000 0000 0046
2. Date and time media file added to WMP (28 bytes after Date and time media file added to WMP (1))
  - a. Date and time media file added takes up eight bytes
3. Date and time media file added to WMP (8 bytes after date and time media file added (2))
  - a. Date and time media file added takes up eight bytes
4. File size in bytes (8 bytes after time media file added to WMP (3))



- a. Actual file size *not* size of disk
  - b. File size takes up three bytes
5. DOS date file was added to WMP (74 bytes after file size in bytes (3))
    - a. DOS date file was added to WMP takes up eight bytes
  6. Name of the audio file (77 bytes after DOS date file was added (5))
  7. Name of audio file in ASCII (44 bytes after DOS date file was added (6))
  8. Full file path and name of audio file (131 bytes after name of audio file in ASCII (7))
  9. The name of the computer which the *CurrentDatabase\_372* resides(58 bytes after Full file path and name of audio file)

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	10	0123456789ABCDEF0	
0x103FF4	0000	0000	0000	0000	0010	2502	7183	05BB	DF	.....\$.qf.»B	
0x104005	0450	003D	0000	0000	0000	0001	1402	0000	00	.P.=.....	
0x104016	0000	C000	0000	0000	0046	4C00	0000	0114	02	..Ä.....Ff.....	1
0x104027	0000	0000	00C0	0000	0000	0000	4683	0120	00	.....Ä.....Ff.....	
0x104038	2000	0000	596C	4062	25EB	CB01	596C	4062	25	...Yl@b*E.Yl@b*	2
0x104049	EBCB	0157	4DE8	7B44	04CA	01F1	6480	0000	00	..E..WMe{D.E.fde...	3
0x10405A	0000	0100	0000	0000	0000	0000	0000	0000	00	.....	4
0x10406B	0029	0114	001F	50E0	4FD0	20EA	3A69	10A2	D8	.)....PäOð é:i.eø	
0x10407C	0800	2B30	309D	1900	2F44	3A5C	0000	0000	00	..+00.../D:\.....	
0x10408D	0000	0000	0000	0000	0000	0000	0000	4C00	31	.....L.l	5
0x10409E	0000	0000	0079	3EDA	9D11	004D	7573	6963	00	...y>Ü...Music.	
0x1040AF	3800	0800	0400	EFBE	543E	B70A	793E	DA9D	2A	8.....i%T>..y>Ü *	
0x1040C0	0000	0025	0000	0000	0001	0000	0000	0000	00	.....\$.....	
0x1040D1	0000	0000	0000	0000	4D00	7500	7300	6900	63	.....M.u.s.i.c	
0x1040E2	0000	0014	00AE	0032	00F1	6480	00EE	3A10	2C	.....ø.2.fde.i.i..	
0x1040F3	2000	4E61	6C69	6D62	612E	6D70	3300	9400	08	..Kalimba.mp3~..	6
0x104104	0004	00EF	BE79	3EDA	9D79	3EDA	9D2A	0000	00	...i%y>Ü y>Ü *	
0x104115	3A8A	0000	0000	0200	0000	0000	0000	0000	42	..Š.....B	7
0x104126	0000	0000	0C4B	0061	006C	0069	006D	0062	00	.....K.a.l.i.m.b.	
0x104137	6100	2E00	6D00	7000	3300	0000	4000	4300	3A	a...m.p.3...@.C.:	
0x104148	005C	0057	0069	006E	0064	006F	0077	0073	00	..W.i.n.d.o.w.s.	
0x104159	5C00	7300	7900	7300	7400	6500	6D00	3300	32	..s.y.s.t.e.m.3.2	
0x10416A	005C	0053	0061	006D	0070	006C	0065	0052	00	..S.a.m.p.l.e.R.	
0x10417B	6500	7300	2E00	6400	6C00	6C00	2C00	2D00	31	e.s...d.l.l.,-1	
0x10418C	0031	0036	0000	001A	0000	0048	0000	001C	00	.1.6.....H.....	
0x10419D	0000	0100	0000	1C00	0000	3200	0000	0000	00	.....2.....	
0x1041AE	0047	0000	0016	0000	0003	0000	00FC	B67F	8C	.G.....üq[]@	
0x1041BF	1000	0000	3530	3047	4200	443A	5C4D	7573	69	....500GB.D:\Mus1	8
0x1041D0	635C	4B61	6C69	6D62	612E	6D70	3300	0028	00	c\Kalimba.mp3~..(.	
0x1041E1	0000	0900	00A0	1C00	0000	3153	5053	E28A	58	.....1SPSäŠX	
0x1041F2	46BC	4C38	43BB	FC13	9326	986D	CE00	0000	00	F%L8C>ü."e~mfi....	
0x104203	0000	0000	6000	0000	0300	00A0	5800	0000	00	.....X.....	
0x104214	0000	0C6E	697A	6D6F	6B00	0000	0000	0000	00	...nizmon.....	9
0x104225	0000	067D	63C3	C7BF	EC4B	8F12	FDAC	F21D	EA	...}cÄÇçik.ý-ö.é	

(Figure 7 – shows the layout of an audio entry available later on the WMP database)  
 The first instance of an audio entry within the WMP database contains the most unique information. Information from the second, third and fourth instances for audio entries repeat what is available within the first as shown above. All information from the first audio entry instance will be extracted and includes.

1. File size
2. Date and time audio file was added to WMP
3. Date and time audio file was last played using WMP
4. Number of plays

5. File name
6. Full file path and name of file
7. Extra details about the media file from the properties of the file

None of the information contained within the second, third and fourth audio entry instances will be extracted, as mentioned above it contains repeated information.

### 2.7.2. Image Layout

The first image entry as with the first audio entry contains the most information; the information available from the image entry is also similar to that of the audio entry.

*(Appendix – Images -Figure 7)*

1. File entry signature
  - a. First audio entries start with the Hex values 2500 0000 88B9 or 2500 0000 08B9, it is not known why there are two sets of file signatures due to the lack of documentation from Microsoft®
2. File size in bytes (16 bytes after file entry signature)
  - a. Actual file size *not* size of disk
  - b. File size takes up three bytes
3. Year the file was created (100 bytes after file entry signature)
  - a. Year file was created takes up five bytes
4. Width of the file in pixels (103 bytes after file entry signature)
  - a. Width of file in pixels takes up six bytes
5. Date and time file was added to WMP (112 bytes after file entry signature)
  - a. Date and time file was added to WMP takes up eight bytes
6. Date and time file was added to WMP (120 bytes after file entry signature)
  - a. Date and time file was added to WMP takes up eight bytes
7. File Name (166 bytes after file entry signature)
8. Full file path including file name
9. Extra details about media file

As with the *Audio Layout* (2.7.1) the second, third and fourth entry are all grouped within close proximity of each other, the number of bytes between the first entry of an audio entry and the second, third and fourth varies, there does not appear to be a noticeable pattern.

(Appendix – Images -Figure 8)

1. File entry signature
  - a. C000 0000 0000 0046 4C
2. Date and time file added to WMP (27 bytes after file entry signature)
  - a. Date and time file added to WMP takes up eight bytes
3. Date and time file added to WMP (35 bytes after file entry signature)
  - a. Date and time file added to WMP takes up eight bytes
4. File size in bytes (51 bytes after file entry signature)
  - a. Actual file size *not* size of disk
  - b. File size in bytes takes up three bytes
5. File Name (168 bytes after file size in bytes (4))
6. DOS date and time file added to WMP (12 bytes after file name (5))
  - a. DOS date and time takes up 7 bytes
7. File name in ASCII (25 bytes after DOS date and time (6))
8. Full file path and name of image file (131 bytes after file name in ASCII (7))
9. The name of the computer which the *CurrentDatabase\_372* resides (58 bytes after full file path and name of image file)

As with the first instance of an audio entry, the first instance of an image entry within the WMP database contains the most unique information. Information shown below will be extracted from the first image entry as shown within *Image Layout* (2.7.2).

1. File size in bytes
2. Date and time audio file was added to WMP
3. File name
4. Full file path and name of file
5. Extra details about the media file from the properties of the file

The first entry within the *Image Layout* (2.7.2) also contains the file size in pixels and the year the file was created, it been decided that both of these details will not be

extracted. Neither the file size in pixels; nor the year the file was created describe information about the type of media viewed using WMP. They merely provide details about it. While the same could be argued for the file size, the size of the file will enable the forensic examiner to recover the file from unallocated clusters if the file has been deleted. The file size allows the forensic examiner to extract the correct amount of bytes from unallocated cluster ensuring the whole file is recovered. As with the audio layout, none of the information contained within the second, third and fourth audio entry instances will be extracted, it contains repeated information as shown within *Image Layout* (2.7.2).

### 2.7.3. Video Layout

The first video entry as with the first audio and video entry contain the most information; the information available from the video entry is also similar to that of the audio and image entry.

(Appendix – *Images – Figure 9*)

1. File signature
  - a. 3A00 0000 FC3F 1209 0E03 B0FC or 10F8 1308 4E03 B0FC, it is not known why there are two sets of file signatures due to the lack of documentation from Microsoft®
2. File size in bytes (89 bytes after file entry signature)
  - a. Actual file size *not* size of disk
  - b. File size takes up four bytes
3. Time when file was added to WMP (133 bytes after file entry signature)
  - a. File size takes up eight bytes
4. Time when file was added to WMP (141 bytes after file entry signature)
  - a. File size takes up eight bytes
5. Times file viewed (191 bytes after file entry signature)
  - a. Times file viewed takes up one byte
  - b. For this to be incremented the “Seek” bar must reach the end
  - c. Skipping to almost the end using the “Seek” bar counts as a viewed
6. File name (265 bytes after file entry signature)
7. Full file path of file including file name

## 8. Extra information

As with the *Audio Layout* (2.7.1) and *Image Layout* (2.7.2) the second, third and fourth entry are all grouped within close proximity of each other, the number of bytes between the first entry of an audio entry and the second, third and fourth varies, there does not appear to be a noticeable pattern.

(Appendix – *Images –Figure 10*)

1. File signature
  - a. 4C00 000 0114 0200
2. Time when file was added to WMP (20 bytes after file entry signature)
  - a. Time when file was added takes up eight bytes
3. Time when file was added to WMP (28 bytes after file entry signature)
  - a. Time when file was added takes up eight bytes
4. File size in bytes (44 bytes after file entry signature)
  - a. Actual file size *not* size of disk
  - b. File size in bytes takes up three bytes
5. DOS date file was added to WMP (141 bytes after file entry signature)
  - a. DOS date file was added to WMP takes up seven bytes
6. DOS file naming convention (209 bytes after file entry signature)
7. ACSII naming convention (44 bytes after DOS file naming convention (6))
8. Full file path including name (51 bytes after ACSII naming convention (7))
9. Computer name (58 bytes after full file path including name (8))

As with the first instance of an audio entry and image entry, the first instance of video entry within the WMP database contains the most unique information. Information shown below will be extracted from the first video entry as shown within *Video Layout* (2.7.3).

1. File size in bytes
2. Date and time when file was added to WMP
3. Number of times viewed
4. File name

5. Full file path and name of file
6. Extra details about the media file from the properties of the file

As with the audio layout and image layout, none of the information contained within the second, third and fourth audio entry instances will be extracted, it contains repeated information as shown within *Video Layout* (2.7.3).

#### 2.7.4. Deleting Media from the Windows Media Player Library

There are several ways in which it is possible for a user to remove media files from WMP. To ensure that WMP does not remove entries from within the WMP database file when a user deletes a media file, all different routes in which a user can delete media from both WMP must be tested. The tests must also cover all media types including; audio, image and video to ensure that accurate results are produced. There are two ways a user can delete a media file from WMP. The first is removing the media file using Windows Explorer. To test that the entry is not removed an audio, video and image file were deleted from within Windows Explorer while WMP was not running and the Recycle Bin emptied. After the removal of the media files WMP was restarted to allow WMP to update the WMP database, after checking the WMP database using an Hex editor the media entries still resided within database file.

The second is to right click the file from within WMP and select “Delete”; the user is then presented with two options. The options are “Delete from library” and “Delete from library and computer”, the latter of the two options just removes the entry from WMP. The former removes it from WMP as well as from Windows and places the media file in the Recycle bin. As with the removal of media from Windows Explorer above all media types must be tested using both options to ensure that accurate results are produced. After the removal of all media types from the WMP library using both “Delete from library” and “Delete from library and computer” the emptying of the Recycle bin and the restarting of WMP to allow it to update the WMP database the file entries remained within the WMP database file. As mentioned above; with the WMP database containing entries of information about files that have been removed from the WMP library, it allows the forensic examiner to see what media the suspect has been viewing even if it has been deleted.

#### 2.8. Overview of WMPLAYER.EXE-xxxxxxx.pf

This section will discuss the background to Windows® Prefetch files and their purpose within the Windows® Operating System. This section will also cover the Windows Media Player (WMP) within Windows® 7 and what information it contains about media viewed using WMP.

##### 2.8.1. Background to Windows Prefetch files

Prefetch files within Windows are designed to speed up the boot process as well as speeding up the loading of programs; it was first introduced in Windows® XP (Microsoft® Corporation, a 2010). Windows® needs to load different segments of the same file into memory at different times which means that a lot of read and writes are needed. To speed up this process Windows® makes a note of the code segments that are loaded and creates a file with information of the most used segments and stores it

in a Prefetch file with the extension “.pf” This allows for less read and writes, therefore it processes to start quicker (Karp,D.A , 2004). Windows® 7 default location for Prefetch files is *root:\Windows\Prefetch*.

### 2.8.2.Windows Media Player Prefetch in Windows 7

When a user runs WMP Prefetch files are created within *root:\Windows\Prefetch\*. The names of the Prefetch file always start with the name of executable in this case “WMPLAYER.EXE” but the ending characters depend on where the file was run from. The ending eight characters are the folder path hashvalues; as long as the WMP executable is within the same folder the ending eight characters will be the same. Below is a list that shows that running WMP from different location causes the files name to change. This list could be argued as the most common ways a user can run WMP. The user could move the executable to anywhere they have permission to write to on the computer and run it causing an almost infinite number of file names.

- Windows Task Bar/Windows Start Menu
  - WMPLAYER.EXE-B0AD61F0.pf
- root:\Program Files\Windows Media Player
  - WMPLAYER.EXE-F4B5869D.pf
- Windows Desktop
  - WMPLAYER.EXE-D7B6CA4C.pf

*Figure 11* - below shows information available at the start of the Windows® Prefetch files.

1. All Prefetch files within Windows® 7 start with 17 00 at file offset 0 and 1.
2. All Prefetch files within Windows® 7 at file offset 4-7 are the same, these Hex values can be used to create a Grep<sup>6</sup> search within EnCase to search through unallocated<sup>7</sup> cluster for deleted Prefetch files.
3. When converted into decimal this shows the size of the file in bytes, this in conjunction with the first few bytes allows the examiner manually extract the file from unallocated clusters.

---

<sup>6</sup>EnCase Grep searches are used as a powerful way to search through data using multiple criteria such as ASCII or Hex values.

<sup>7</sup> Unallocated clusters are clusters of data that have been deleted and not yet overwritten so still contain information.



4. This shows the name of the file that the Prefetch relates to, this is useful if the file is either being recovered from unallocated clusters or if the files have been renamed within Windows®.
5. This shows the last eight characters of the Prefetch file name, the example below F061 ADB0 would become ADB0 61F0.

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF		
1	0x00000	1700	0000	5343	4341	1100	0000	6C8C	0200	....SCCA....lE..	3
	0x00010	5700	4D00	5000	4C00	4100	5900	4500	5200	W.M.P.L.A.Y.E.R.	
	0x00020	2E00	4500	5800	4500	0000	0000	0000	0000	..E.X.E.....	4
	0x00030	4D00	0000	80FA	FFFF	0000	0000	0000	0000	M...€úÿÿ.....	5
	0x00040	0000	0000	0000	0000	1E0B	0A03	F061	ADB0	.....δa-°	
	0x00050	0000	0000	F000	0000	F900	0000	1020	0000	....δ....ù....	

(Figure 11 – shows information located at the start of Windows® Prefetch files)

Records within the *WMPLAYER.EXE-xxxxxxx.pf*(xxxxxxx denotes the ending eight characters) file are not always the same length due to the location and name of the file that is being called upon; because of this it would not be possible to create a software tool that pulls out prefixed lengths of the file. After investigating the *WMPLAYER.EXE-xxxxxxx.pf* it is clear that each entry within the file begins with .\D.E.V.I.C.E.\, with each entry starting with the same Hex values it will be possible to create a software tool that parses information until it finds a valid entry and separates them from the next entry and extracts that information to be viewed by forensics examiners.

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF	
0x1ABD0	4900	0000	5C00	4400	4500	5600	4900	4300	I...\.D.E.V.I.C.	1
0x1ABE0	4500	5C00	4800	4100	5200	4400	4400	4900	E...\.H.A.R.D.D.I.	
0x1ABF0	5300	4B00	5600	4F00	4C00	5500	4D00	4500	S.K.V.O.L.U.M.E.	
0x1AC00	3400	5C00	5500	5300	4500	5200	5300	5C00	4...\.U.S.E.R.S.\.	
0x1AC10	4A00	4100	4D00	4500	5300	5C00	4D00	5500	J.A.M.E.S.\.M.U.	
0x1AC20	5300	4900	4300	5C00	5A00	5C00	4B00	4100	S.I.C.\.Z.\.K.A.	
0x1AC30	4C00	4900	4D00	4200	4100	2E00	4D00	5000	L.I.M.B.A...M.P.	2
0x1AC40	3300	0000	5C00	4400	4500	5600	4900	4300	3...\.D.E.V.I.C.	

(Figure 12– shows the start on an entry within the WMP Prefetch file)

The entry within the *WMPLAYER.EXE-xxxxxxx.pf* allows the examiner to see the full directory of the media viewed using WMP, including the volume the media is located on. The Prefetch file is essential in the investigation of the use of WMP; while the WMP database records a lot more meta data about the file played the file needs to be imported for WMP to record the information, whereas the Prefetch files does not. The media does *not* have to have been played for it to be recorded within the Prefetch file, if a user selects 10 files to be viewed using WMP they are potentially recorded within the Prefetch file. It is not currently known how Windows® chooses to which entries to add to the Prefetch files as there is not documentation released from

Microsoft® on this area. Whilst it could be argued that the user never necessarily viewed the media it shows intent to do so and ultimately the user was aware of what they were doing.

While the Prefetch file *WMPLAYER.EXE- xxxxxxxx.pf* stores information about media viewed using WMP that might not be recorded with the WMP database file, it is shared by all users of the computer as it is located within *root\Windows\Prefetch* in Windows® 7. To test this a file was played using WMP on one account which was then logged out of, after logging into another account the Prefetch file contained an entry for the file played on the first account. While this is not a problem if there is only one account on the computer (providing that no accounts have been deleted or *guest* account enabled), if there is more than one account the forensic investigator will need to determine which account caused the entry to reside within the Prefetch file. The forensic examiner will be able to do this by using the file directory stored within the Prefetch file.

Unlike the WMP database file the WMP Prefetch is sometimes deleted through Windows® updates released every Tuesday, programs that are designed to speed up the performance of computers or just purged “every three to twenty eight days” (Hay, S.A 2010). There are also a number of websites which suggest that a user cleans out their Prefetch folder (Cool Buster, 2010). Even if the Prefetch file for WMP has been deleted through the user’s unknowing or knowing actions in the attempt to speed up their machine or destroy evidence. The forensic examiner may be able to recover the file from unallocated clusters by searching for the Grep expression in EnCase as noted above. One of the advantages of the Prefetch file is that it records the full directory of the file; one of the benefits of this is if a user is using a hidden volume such as those that can be created using TrueCrypt the directory will be recorded. Since TrueCrypt has the ability to hide directories from view, finding the path listed in a Prefetch file can provide a data source that might not otherwise be identified leading to further investigation.

## 2.9.Current solutions

### 2.9.1.Windows Media Player Database

After conducting research it has become apparent that multiple people have been interested in the extraction of information from within the WMP database (Forensic Focus, a 2006) file. This has been discussed several times upon a forum *Forensic Focus* designed for computer forensics personnel (Forensic Focus, b 2010). This forum was recommended as a good source of information by Detective Allan Hay from the Northumbria High Tech Crime Unit. While the posts are dated 2006 they do discuss the lack of tools for the extraction of information about media viewed using WMP; upon the Windows® Operating System.

One current solution to extracting information stored within the Windows Media Player database file is a program called Windows Media Player Database Extractor (WMDB Extractor) by a company called Filesig Software Solutions (Filesig Software Solutions, 2010). The current solution provided by Filesig Software Solutions is designed to extract the information from within the *CurrentDatabase\_360.wmdb*

which is the database created by WMP player version 11 (Microsoft® Corporation, g 2009). It has not been possible to obtain a copy of WMDB Extractor so it has not been possible to test its effectiveness on the current WMP database.

Detective Allan Hay is currently investigating the WMP database file in Windows® 7 and is creating a software tool to extract information from within it. After several searches it has not been possible to find any other information with regards to this.

### 2.9.2.Windows Prefetch

Whilst there has been no program designed specifically to extract information from Windows Prefetch files about media viewed using WMP, the program *Advanced Prefetch File Analyser* (APFA) version 2.1 is able to extract entries from Prefetch files which can include entries about media viewed using WMP (Hay, S.A, 2010). While AFPA reports on what media has been viewed using WMP within the *WMPLAYER.EXE-xxxxxxx.pf* it also reports other files which could be considered irrelevant such as Dynamic Link Libraries (DLL) and Windows Executable (EXE) files etc. Without the ability to search by type of file run the forensic examiner will have to manually go through possibly thousands of entries to look for relevant entries.

A software tool called *Prefetch Information* created by Mark McKinnon (Prefetch Information, 2009) is available as a free download. The description describes “What it will do is parse the Prefetch file giving you the standard information that other programs have given i.e. embedded date, number of time run and executable name plus a list of directories and files that are/have been loaded.”. After the testing of this program it is unable to examine the *WMPLAYER.EXE-xxxxxxx.pf* which contains information about media that has been viewed using WMP within Windows® 7.

### 2.10.Possible Approaches

The research conducted so far has shown that; the issue of extracting information about media viewed using WMP within Windows® 7 as well as other versions of Windows®, has gone largely unnoticed within the forensic community. The need for a solution is shown by discussions upon forensic forums (Forensic Focus, a 2006). Below will discuss the possible approaches that can be undertaken in aim of providing a solution for the extraction of media viewed using WMP within Windows® 7.

Firstly it must be ascertained how the forensic examiner will use the software to be constructed to perform an analysis of both the WMP database and WMP Prefetch. If a solution is devised that extracts information from both the WMP database and WMP Prefetch files, but is likely to be unused by an examiner the creation of a solution will have been worthless. There are two ways a forensic examiner could conduct an analysis of the WMP database and WMP Prefetch files, a “live” analysis and a “static” analysis. Both approaches have advantages and disadvantages. The former of the two; live analysis, requires that a suspects machine be seized in an on state with the ability to interact with the operating system i.e. not locked and password protected.

Performing a live analysis has many advantages including, capturing of data stored within Random Access Memory (RAM) before power is removed and capturing data while unencrypted. The advantage of a live analysis within the scope of this project is

the returning of results about media viewed using WMP within minutes of a computer being seized. Once a machine has been seized in an on state, the forensic examiner would be able to conduct an analysis of the WMP database and WMP Prefetch files to determine what media a suspect has been viewing. By reviewing the information returned from the analysis, the forensic examiner could determine whether to seize the machine to conduct a further analysis or not. The ability to determine whether to seize a suspects machine has the ability to reduce the “ten month national backlog” (Kennedy, 2009) of forensic cases. If an examiner were to use a solution for the extraction of media viewed using WMP upon a suspects machine, and after using the solution the suspects machine was *not* seized, but would otherwise have been if no such solution existed, potentially tens of hours of examination time would have been saved. This highlights a solution to the extraction of information about media viewed; using WMP would be beneficial to potentially reducing the case load computer forensic examiners face. While results produced about media viewed using WMP do *not* prove or disprove the illegal viewing of media upon a suspects machine, results will help prioritise cases.

Potentially the main disadvantage to performing a live analysis is the breaking of evidential integrity. The performing of a live analysis breaks the first ACPO principle.

**Guideline 1:** “No action taken by law enforcement agencies or their agents should change data held on a computer or storage media which may subsequently be relied upon in court.” (ACPO, 2010)

While performing an analysis of the WMP database and WMP Prefetch breaks the first ACPO guideline, the second guideline allows for this.

**Guideline 2** “In exceptional circumstances, where a person finds it necessary to access original data held on a computer or on storage media, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions.” (ACPO, 2010)

As with performing a live analysis, performing a static analysis has both advantages and disadvantages. The advantage of performing a static analysis upon a suspect’s machine is the maintaining of evidential integrity. When the suspects machine is seized the power is removed, this ensures that no changes are made by the forensic examiner. The forensic examiner then creates a bit for bit images of the suspects’ hard drive. The analysis of the WMP database and WMP Prefetch files are then conducted upon the identical copy of the suspects’ hard drive.

Performing a static analysis of a suspect’s machine has the disadvantage of losing information stored within the RAM as well as any information that was unencrypted prior to the power being removed. The disadvantage of performing a static analysis within the scope of this project is the loss of time. Once the machine has been seized as mentioned above, a bit for bit copy of the hard drive must first be created. Depending on the size of the suspect’s hard drive, depends on the amount of time it takes to make an identical copy of it. With hard drive capacity increasing the amount of time it takes to make a bit for bit copy will also increase. The acquisition of an

identical image of the suspects hard drive has the potential to take several days depending on the immediate success of the imaging process, only once an identical copy of the suspects hard drive has been obtained can an analysis of the WMP database and WMP Prefetch be conducted. Once the analysis of the WMP database and WMP Prefetch is complete, it may be found that no illegal media has been viewed using WMP.

To enable forensic examiners to reduce the backlog of cases and have the greatest flexibility, a solution that allows both a “live” and “static” analysis of the WMP database and WMP Prefetch will be constructed. By providing the forensic examiner with the ability to perform both a live and static analysis, it would enable the prioritization of cases by performing a live analysis to determine if further investigative work is required. Allowing the forensic examiner to conduct a static analysis would aid where there is need for evidential integrity or the examiner is unable to justify the breaking of the first ACPO guideline.

Now it has been ascertained; that the solution to the extraction of information about media viewed using WMP, within Windows® 7 needs to be able to perform both live and static analysis. The solution must have the ability to execute from a Universal Serial Bus (USB) storage device. Enable the solution to perform an analysis of from a USB storage device is similar to CheckStick and RAM (Hay, A 2010).

#### 2.10.1.Cluster Analysis

Cluster analysis or ‘clustering’ is used in many fields including data mining and pattern recognition. A cluster is defined as a collection of objects relatively similar to one another and relatively dissimilar to other clusters (Chen, Y 2006). Similar clusters refer to the known file entry headers discovered through research into WMP database and WMP Prefetch files as show in the *Overview of CurrentDatabase\_372.wmdb* (2.7) and *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8).

Cluster analysis within the scope of this project will be used in discovering the similarities in data within both the WMP database and WMP Prefetch. The hexadecimal clusters of data or file entry headers laid out in both the *Overview of CurrentDatabase\_372.wmdb* (2.7) and *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8), will allow the use of cluster analysis to be employed as a means of searching for potential known data sets.

#### 2.11.Software Requirements and Justification

Below are the requirements that have been laid out to which the software must adhere to. These requirements will be used throughout the report to help determine if project has been successful and to aid in the construction. The requirements will be used to compare the software’s functionality against the requirements set below.

1. Extract information about media viewed upon Windows 7
  - a. Extract information from the *CurrentDatabase\_372.wmdb*
  - b. Extract information from *WMPLAYER.EXE-xxxxxxx.pf* files

2. Ensure the evidential integrity is maintained when extracting information
  - a. Extract exact copies of entries from the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
  - b. Ensure no alterations are made to either the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
3. Produce a report for easy analysis
4. Have Graphical User Interface (GUI) to allow for ease of use

Each of the software's requirements has been outlined above. Below will discuss each of the requirements individually, and the rationale behind them.

#### 2.11.1. Extract information about media viewed upon Windows 7

As discussed within *The Significance of Windows Media Player in a Forensic Investigation* (2.2) the purpose of this project is to allow the forensic examiner to view media viewed using WMP upon Windows® 7. To allow the forensic examiner to view what the suspect has been viewing, information needs to be extracted from both the WMP database and the WMP Prefetch. Information about media viewed using WMP stored within the WMP database is arguably more important compared to information stored within the WMP Prefetch. The information stored within the WMP database holds information such as time added to WMP, last viewed and number of times played. With the WMP database having more information available to the forensic examiner compared to the WMP Prefetch, it will allow them to build a better picture of what the suspect(s) have been doing. Ensuring that information extracted from the WMP database is key in providing the forensic examiner with as much information as possible about media viewed using WMP. Information about media viewed using WMP stored within the WMP Prefetch while not as detailed due to only containing the full file path including name as shown within *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8), is still important to help the forensic examiner build a picture of what the suspect(s) has been doing. While the WMP Prefetch files do not store as much information about media viewed using WMP as the WMP database, it can contain information that is potentially not available within the WMP database as shown within *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8). For this reason the information stored within the WMP Prefetch files about media viewed using WMP is essential. Through providing information potentially not available within the WMP database it minimises that key information missed.

#### 2.11.2. Ensure the evidential integrity is maintained when extracting information

Evidential integrity is integral to the results produced by the software about media viewed using WMP upon Windows® 7. Evidential integrity is the process of ensuring that the information presented is an exact copy of the original with no changes made. In the case of the software to be constructed, it must be able to extract entries from



within both the WMP database and WMP Prefetch files without altering the data in anyway. If entries from within either file(s) are changed in anyway, evidential integrity will have been broken. If evidential integrity is broken, it can be argued that the results are unreliable due to them not being the same as the original. The software to be constructed must maintain evidential integrity when extracting entries from within the WMP database and WMP Prefetch files to ensure that their accuracy is not called into question.

Not only must information extracted maintain evidential integrity by ensuring they are an exact replica of the results within both the WMP database and WMP Prefetch, the evidential integrity of both files must also be maintained. While extracting information from both the WMP database and WMP Prefetch files no alterations must be made to the files. Any alterations to the WMP database or WMP Prefetch files will break evidential integrity. As with extracting entries from within the WMP database and WMP Prefetch, if evidential integrity is broken the results that are produced can be argued unreliable due the files not being the same as prior to extraction. The software to be constructed must maintain evidential integrity by *not* altering the WMP database or WMP Prefetch; this will ensure that the accuracy of the results is not called into question, as the files will be the same as if the extraction had not been completed. Ensuring the evidential integrity of both entries extracted from the WMP database and WMP Prefetch files, as well as the files themselves ensures the compliance with the first ACPO) guideline as discussed within *Possible Approaches* (2.10).

Within *Possible Approaches* (2.10), it was decided that software tool to be constructed should be able to be run in live environment. To enable the software tool to run within a live environment it will have the ability to run from a USB memory stick. It is important to understand the ramifications of the use of a USB memory stick. When external storage devices are plugged into a Windows® based Operating System information is recorded within several locations relating to that device. Within Windows® 7 information is recorded about external storage devices within in the Registry. Within the registry key

HKEY\_LOCAL\_MACHINE\SYSTEM\MountedDevices, information regarding the USB's serial number and drive letter used to mount the device are recorded. With information within the Registry being altered due to the use of a USB memory stick, the first ACPO guideline will have been broken. The second ACPO guideline allows for the first to be broken so long as the forensic examiner can account for the changes as discussed within *Possible Approaches* (2.10).

To ensure that the forensic examiner can account for the changes made to the suspect's computer, the details of the USB memory stick such as the serial number should be noted down. By knowing what entries relate to the forensic examiners USB storage device, a process of elimination can be used to show what was and was not altered by the examiner.

Civil cases have the possibility of going criminal; if the ACPO guidelines are always followed the evidence used within the civil case could be used during the criminal

case. Not only could the evidence extracted from the WMP database and WMP Prefetch be used in court that was used during the civil investigation, it would also allow the WMP database and WMP Prefetch files to be used during the criminal case as their evidential integrity will have been maintained during the civil case.

The changes made to a suspects' computer could be considered minimal to the benefit of the ability to perform a live analysis. As noted within *Possible Approaches* (2.10) the machine must be powered on and in an unlocked state. If the machine is powered off, the forensic examiner is able to turn the machine on as long as the process is documented (ACPO, 2010). Whether or not the forensic examiner turns on the machine should be down to them to judge each case on its own merits.

To ensure further evidential integrity the software to be developed should only be able to interact with both the WMP database and WMP Prefetch. The software to be developed should not interact with the operating system another that what is necessary to run the software, results from the computer that is being analysed should be saved to the location for example. This not only ensures that there is a minimal footprint from the analysis of both the WMP database and WMP Prefetch but that evidence collected is not left behind. To counter this issue the software should only be able to save reports from the device on which it is running.

#### 2.11.3. Produce a report for easy analysis

A key component in a forensic tool is the presentation of evidence. The better presented the evidence, the easier it will allow the forensic examiner to determine in the context of this project what the suspects has been viewing using WMP.

To allow the forensic examiner to review the information about media viewed using WMP upon Windows® 7 retrieved from both the WMP database and WMP Prefetch, the information must be stored in a permanent state. With the possibility of hundreds or thousands of entries retrieved from WMP database and WMP Prefetch information will need to be well presented. Well-presented information within the report will allow the forensic examiner to distinguish between records better as well as review the contents of each entry. If the software produces a report that is difficult to read; such that the forensic examiner cannot differentiate between records, it may cause confusion about the information presented.

#### 2.11.4. Have Graphical User Interface (GUI) to allow for ease of use

While the software is a proof of concept, having a GUI has advantages over using a Command Line Interface (CLI). The first advantages with using a GUI within this project is the ability for less experienced or non-forensic officers to perform an analysis of the WMP database and WMP Prefetch. By creating a GUI that is intuitive to use, such that it mimics other software in its design, its user base increases beyond the highly trained forensic examiner. If non-forensic personnel were to use the software solution, such as police officers with little to no forensic training it has the potential to reduce the national back as mentioned within *Possible Approach* (2.10). Police officers could determine the priority of a suspect's computer depending on the results produced from the analysis of the WMP Extractor or WMP Prefetch, they could determine if a machine should be seized or not. If a police officer has no



way of knowing if the machine has been used to view illegal media it must be seized to ensure potential cases are not missed. If a suspects machine is seized that is later found to not contain any illegal media, this will further add to the national backlog by taking examination hours away from other cases. The use of a CLI would reduce the user base of the software solution as the technical knowledge required to operate would be outside of the scope of an inexperienced user.

As well as increasing the software solutions user base, the use of a GUI within this project allows for a controlled use of the software. Depending on the stage of analysis of the WMP database and WMP Prefetch files, only certain options would be accessible to the forensic examiner. While the software is in the process of searching for entries within both the WMP database and WMP Prefetch, the option to exit or select different files would be disabled; preventing the potential failure of an analysis.

If a forensic examiner is asked for the directory of either the WMP database or WMP Prefetch through the use of a CLI and the wrong directory is entered, an analysis may not be conducted as the files may not exist within the directory given. This may lead the forensic examiner to believe that the files do not exist upon the computer, and not proceed any further. The use of a GUI would allow the forensic examiner to select the directory where the WMP database or WMP Prefetch files are located, while the examiner would still be able to select an incorrect directory, typos won't be the cause of this.

## 2.12. Identification of Tools and Techniques

Having ascertained the layout of the WMP database, appropriate tools and techniques need to be considered to enable to the best approach of the extraction of media viewed using WMP within Windows® 7. Below will discuss the different tools, methods and techniques for use in both the design and implementation of the software.

To enable the construction of the software a suitable programming language needs to be determined. To determine a suitable programming language, both the file entry headers and requirements of the software must be taken into account. By taking into account the file entry headers and requirements when choosing a programming language it will ensure that the programming language is capable of providing the functionality that is required.

The file entry headers found within the WMP database and WMP Prefetch as shown within *Overview of CurrentDatabase\_372.wmdb* (2.7) and *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8) require the ability to read in files using hexadecimal. Variations of the C programming language are able to handle the use of hexadecimal, such as C# and Java. Both of these languages have a built Hex class that enables the reading of hexadecimal. While the author has knowledge within both languages, the author has a greater understanding within Java; and so has been chosen as the programming language. If the software were to be used upon a live machine, it would require the Java Runtime Environment (JRE). If the suspects machine does not have the JRE installed the analysis can still be conducted through a static analysis.

As mentioned above the programming language chosen must be able to also meet the requirements of the software. The programming language needs to be able to both store information extracted from the WMP database and WMP Prefetch files and, have the ability to create a GUI. Java fulfils both these requirements with the ability to store data in multiple formats such as text file, Excel files and database, Java is also able to create a GUI through the use of the *Swing* library.

Now a suitable programming language has been determined the need for a suitable design methodology is also required. With Java as the programming language the design methodology will be the Unified Modelling Language (UML). The design of the software is discussed within *Design Overview* (3.2).

With a Java chosen as the programming language to be used in the construction of the software application for the extraction of information about media viewed using Windows® 7, a suitable Independent Development Environment (IDE) must be chosen. There are many IDE's that are capable of support Java such as Eclipse (Eclipse, 2011) and NetBeans (NetBeans, 2011), the chosen IDE within this project is JCreator (JCreator, 2011). JCreator has been chosen due to its familiarity with the author throughout programming at university.

### 2.13. Analysis Overview

As determined from the research conducted as part of the analysis section, the WMP database file as well as other areas, holds significant information about media viewed using WMP upon Windows® 7. There is also a current lack of tools available for forensic examiners for the extraction of information about media viewed using WMP within Windows® 7. With this in mind it justifies a need for a solution to be implemented, the research conducted will ensure suitable tools and techniques will be used for the project. The software tool to be created will be referred to as WMP Extractor as is expected to meet the following requirements:

1. Extract information about media viewed upon Windows 7
  - a. Extract information from the *CurrentDatabase\_372.wmdb*
  - b. Extract information from *WMPLAYER.EXE-xxxxxxx.pf* files
2. Ensure the evidential integrity is maintained when extracting information
  - a. Extract exact copies of entries from the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
  - b. Ensure no alterations are made to either the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
3. Produce a report for easy analysis
4. Have Graphical User Interface (GUI) to allow for ease of use



### **3. Synthesis**

The synthesis section will cover the design, implementation and testing of WMP Extractor. This section will also cover any problems that arise during the implementation of the WMP Extractor, as well as how the problems were overcome and any implications these had on the project as a whole.

### 3.1. The problem

As discussed within the analysis, there is currently limited options available to computer forensic examiners for the extraction of media viewed using Windows Media Player (WMP). On investigation undertaken by the author, there is no known current method available for the extraction of this information from Windows® 7. Having established the need for a solution to this problem WMP Extractor was proposed. For this project to be successful WMP Extractor must meet the requirements laid out in *Software Requirements and Justification* (2.11). The use of the known clusters prior to data within both the WMP database and WMP Prefetch as described within both the *Overview of CurrentDatabase\_372.wmdb* (2.7) and *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8), will be used to construct a search algorithm to aid in the extraction of information about media viewed using WMP within Windows® 7.

Some of the problems this project needs to overcome are:.

- The WMP database and WMP Prefetch are fragmented throughout the Operating System (OS)
  - The WMP database and WMP Prefetch are stored in two locations
    - ❓ *root:\Users\%USERS%\AppData\Local\Microsoft\Media Player*
    - ❓ *root:\Windows\Prefetch*
- Media entries are fragmented throughout WMP database and Prefetch files
  - Entries within both files are *not* sequential
  - Both files contain other irrelevant information in the context of this project
- Information within the WMP database and WMP Prefetch is stored in a proprietary format
  - No current information available from Microsoft®
- On investigation by the author there is no known current solution
  - This project is unable to build upon current techniques as none are known

WMP Extractor will attempt to overcome these issues and in the following sections a design, implementation and testing strategy as defined by the requirements specification shown within *Software Requirements and Justification* (2.11) will be discussed.

### 3.2.Design Overview

To enable a successful implementation of WMP Extractor a suitable design process must be chosen. As discussed within *Identification of Tools and Techniques* (2.11) Java was chosen as the programming language of choice and thus the Unified Modelling Language (UML) will be used.

By using the requirements outlined within *Software Requirements and Justification* (2.11) shown below a design can be constructed to ensure that WMP Extractor meets all of them.

1. Extract information about media viewed upon Windows 7
  - a. Extract information from the *CurrentDatabase\_372.wmdb*
  - b. Extract information from *WMPLAYER.EXE-xxxxxxx.pf* files
2. Ensure the evidential integrity is maintained when extracting information
  - a. Extract exact copies of entries from the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
  - b. Ensure no alterations are made to either the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
3. Produce a report for easy analysis
4. Have Graphical User Interface (GUI) to allow for ease of use

While a design is a good guide on how a program should be constructed, it is just that; a guide. The author envisages problems that may result in changes to the design during the implementation; these will be documented and discussed later on.

As discussed within *Identification of Tools and Techniques* (2.12) Java is the chosen language that WMP Extractor will be constructed in. One of the main reasons Java was chosen was because it is an Object Oriented language. Object Oriented languages also allow for the better maintenance of code with the ability to change the objects behaviour should the requirements of the program change throughout time. With this in mind the design process should take into account the use of objects to speed up the implementation process by allowing for the re-use of code as well as code that is easier to maintain. Another reason for the decision to use Java was due to its excellent support of the Hex class as documented within *Identification of Tools and Techniques* (2.12). This will allow for WMP Extractor to read in both the WMP database and WMP Prefetch in Hex, and scan for the clusters that are known to exist prior to entry information existing within the files.

WMP Extractor will have several *classes*; each *class* will have a specific function. Using separate *classes* allows for an easier management of code by separating different functions of WMP Extractor. There will be one master class known as a *main class*. The *main class* will control the program calling upon other *classes* to perform the actions needed to extract the information required to fulfil WMP Extractors requirements as detailed within *Software Requirements and Justification* (2.11).

### 3.3.Unified Modelling Language (UML)

To be able to design WMP Extractor we must have some way to convey how the program will be constructed using descriptions and visual representations. As discussed within *Identification of Tools and Techniques* (2.12), UML will be used.

#### 3.3.1.What is UML?

UML is known as Unified Modelling Language and is used in the design process of Object Orientated programs such as Java in which this project is based.

#### 3.3.2.Reason for using UML

UML offers a rich notation to model software systems and understand it from different viewpoints. Some of the main advantages of UML are below.

1. It is a formal language
  - Every part of UML has a strongly defined meaning allowing for software engineers to know what each part of the program is designed to do.
  - The entire language is made up of straightforward notations
2. It is comprehensive
  - UML is able to describe all aspects of a project
  - UML is able to be used in all aspects of the design
  - UML can be used in the creation of a test strategy as all areas of the software can be designed using it
3. It is the standard
  - UML is an open standard created by Object Management Group (OMG) (Object Management Group, 2011)
  - UML's no proprietary standard avoids vendor lock-in which avoids one company dictating how the standard is set
4. It is built upon lessons learned

- UML has been developed over several years which has refined its effectiveness by implementing changes to resolve issues
- UML's development over several years has provided a solid foundation for its robustness

5. It is universal

- UML can be used to design software in any Object Oriented programming language

### 3.3.3.Chosen UML diagrams

- Use Case Diagram

The Use Case Diagram was chosen because of its ability to capture the requirements of the system. A Use Case Diagram focuses on the behaviour of the system from an external point of view, it uses an actor (a forensic examiner in a real world situation of WMP Extractor) to show the interaction between them and the system (Bruegge, H. Dutoit, A,H, 2009).

- Class Diagrams

The Class Diagram was chosen as it will allow the author to show which classes interact with each other by use of visual representation. The visual representation also shows the inner workings of WMP Extractor. Class diagrams are used to describe the structure of the system to be created; in this case it is the WMP Extractor. A class diagram visually represents all classes of a system and their interrelationships (including inheritance, aggregation, and association) and the operations and attributes of the classes (Sobh, T, 2010).

The Use Case Diagram, Class Diagram and Sequence Diagram have been chosen as they allow the visual representation of different aspects of WMP Extractor, this will allow people with less technical knowledge to understand the design of WMP Extractor and how it works.

The Use Case Diagram will show:

- The different functions of WMP Extractor
- How the actor (forensic examiner) interacts with different functions of the WMP Extractor
- The outcomes from the interactions from the actor

The Class Diagrams will help create a detailed testing strategy by showing how different classes interact with each other, then comparing it to the outcome to ensure every class performs as expected. The Class Diagram will also help debug any problems that arise as it will show which classes interact with each other; this can then be used to narrow down where the problem is occurring.



### 3.3.4. Architecture of WMP Extractor

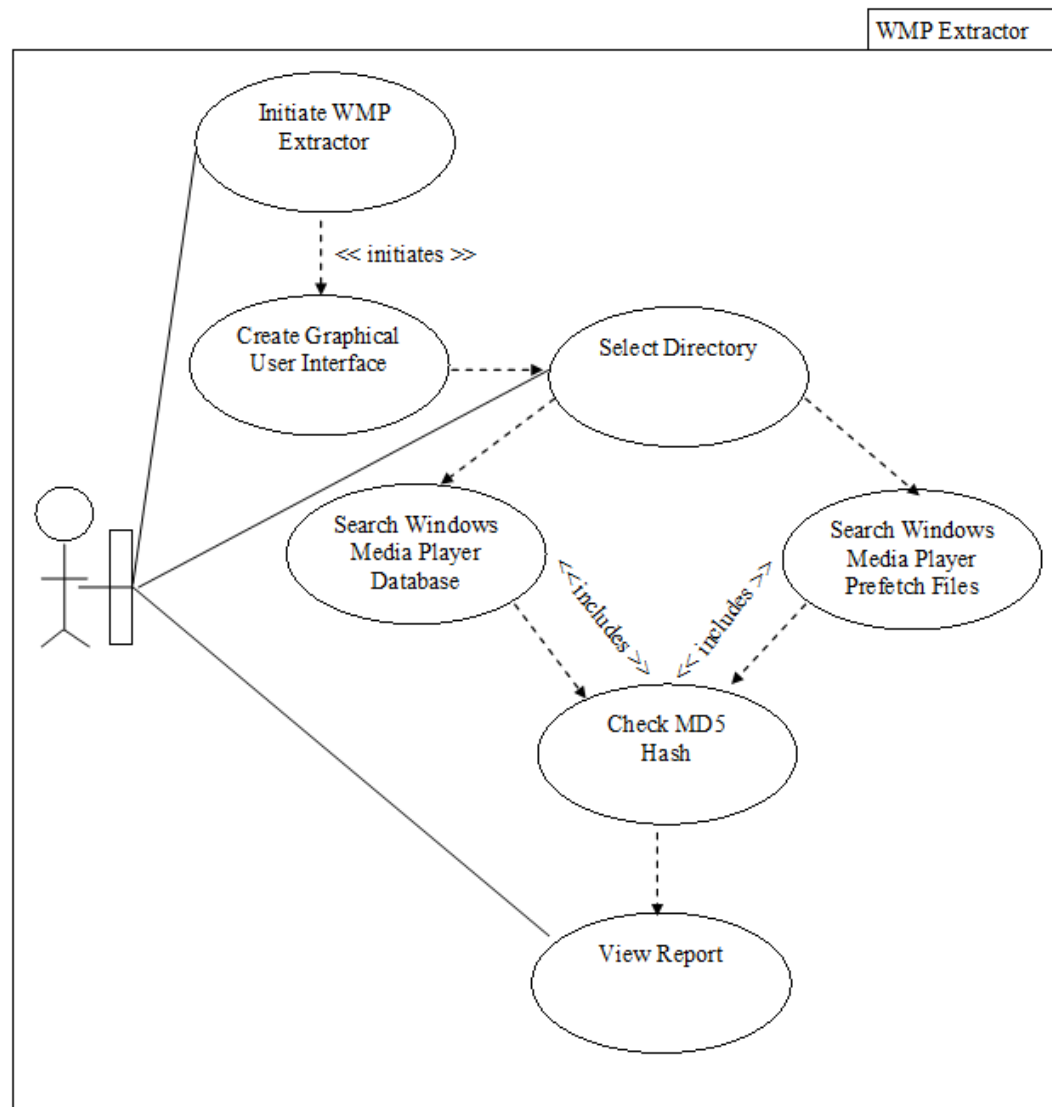
The product requirements described within *Analysis Overview* (2.13) require WMP Extractor to be able to:

1. Extract information about media viewed upon Windows 7
  - a. Extract information from the *CurrentDatabase\_372.wmdb*
  - b. Extract information from *WMPLAYER.EXE-xxxxxxx.pf* files
2. Ensure the evidential integrity is maintained when extracting information
  - a. Extract exact copies of entries from the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
  - b. Ensure no alterations are made to either the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
3. Produce a report for easy analysis
4. Have Graphical User Interface (GUI) to allow for ease of use

While WMP could use a single class that contains all of the code needed to achieve the requirements above this is not best practice. WMP Extractor will be built up using several class files. Each class will have a specific purpose but combined the overall goal of meeting the requirements above. Using separate class files allows for the easy maintenance of code.

### 3.3.5. Use Case Diagram

Figure 12 below shows Use Case Diagram for WMP Extractor.



(Figure 12)

WMP Extractor will be constructed using several class files, below discusses the main aims of each *class*.

WMPExtractor – class

- InitiatesGraphicalUserInterface class

GraphicalUserInterface – class

- Receives an examiner name and case number from the user
- Requests the system date and time from DateAndTime

- Creates the graphical user interface
- Allows the user to select the folder location
  - User is able to default location if running on a live machine
  - User is able to select a specific folder if files have been extracted to another location
- Sends folder location selected to WMDBExtractor

WMDBFileSignatures- class

- Sends headings to OutPutToFile to be outputted into the report
- Sends file signatures to WMDBExtractor

WMDBExtractor - class

- Receives file entry information from WMDBFileSignatures
- Receives folder location from GraphicalUserInterface
- Searches for file entries within the *CurrentDatabase\_372.wmdb*
- Extracts entries in Hex and passes data to HexConverter
- Extracts entries in and passes data to OutputToFile

PrefetchExtensions – class

- Sends headings to OutPutToFile to be outputted into the report
- Sends file signatures to PrefetchExtensions

PrefetchSearch- class

- *Receives* file entry information from PrefetchExtensions
- Receives folder location from GraphicalUserInterface
- Searches for file entries within the *Prefetch* files
- Extracts entries in Hex and passes data to HexConverter

DateAndTime – class

- Receives hex from WMDBExtractor
- Converts Windows® UTC time and outputs to OutputToFile

HexConverter – class

- Converts Hex received from WMDBExtractor and PrefetchSearch into ASCII
- Outputs converted hex to OutputToFile

HashValue – class

- Receives file directory with file name from WMDBExtractor and PrefetchSearch
- Calculates MD5 hash value and outputs to OutputToFile

OutputToFile - class

- Receives data from WMPDBExtractor and HexConverter
- Creates a .tsv file to be read by Excel
- Outputs data to .tsv file

### 3.3.6. Use Case Description

The below (UC Description: Extract Audio from WMP Database) is a class from WMP Extractor and describes the extraction of audio entries from within the WMP database file. The remaining Use Case Descriptions are located within *Appendix -Use Case Description (3.3.6)*.

Use Case	Extract Media Entries From WMP Database
Summary	WMPDBExtractor scans the WMP database for media entries and extracts them out and stores them in the report.
Actor	User
Trigger	WMP Extractor finds the WMP database via the default file path or a manually selected file path
Primary Scenario	<ol style="list-style-type: none"> <li>1. WMP Extractor find the WMP database via the given path</li> <li>2. MD5 hash value checked using HashValue</li> <li>3. WMPDBExtractor receives file information from WMPDBFileSignatures</li> <li>4. WMPDBExtractor searches for known file signature</li> </ol>
Alternative Scenario	<ol style="list-style-type: none"> <li>1. WMP Extractor is unable to find any media entries</li> </ol>
Exceptional Scenario	<ol style="list-style-type: none"> <li>1. WMP Extractor is unable to find the WMP database</li> </ol>
Pre-Conditions	WMP Extractor is running
Post-Conditions	Media entries copied out of database and stored within report
Assumptions	Valid name, case number/name and file name entered

(UC Description: Extract Media Entries from WMP Database)

### 3.3.7. Pseudo Code

Pseudo code is used in the development of software but is not a programming language itself; but similar to everyday English. It allows for the flow of the program to be documented in an easy to read format by laying out the order in which actions occur. A well-constructed pseudo program helps in the implementation by replacing statements with program statements e.g. (Deitel, P.Deitel, H, 2009).

### Pseudo Code

1. Count to ten

### Java Code

```
for (int i = 0; i < 10; i ++){  
}
```

Below is the Pseudo Code for WMDBExtractor.

1. Receive file signature from WMDBFileSignatures
2. Receive file directory from GraphicalUserInterface
3. Check to make sure `CurrentDatabase.wmdbexists` within received directory
  1. If `CurrentDatabase.wmdb` does not exist inform user, run `PrefetchSearch`
4. Read in file using Hex class looking for file entry received from `PrefetchExtrnsions`
5. Extract entries in Hex, pass to `HexConverter`
6. Repeat steps 4 and 5 until every byte has been read in

The reaming Pseudo Code is located within the *Appendix – Pseudo Code* (3.3.7).

### 3.3.8.Class Diagrams

Below is a class diagram for WMDBExtractor.java. The diagram shows the variable used within the class denoted by an – symbol, the methods are denoted by a + symbol.



<ul style="list-style-type: none"> <li>- String filePath</li> <li>- String fileHeaderOne</li> <li>- String fileHeaderTwo</li> <li>- String fileHeaderThree</li> <li>- String fileHeaderRemaining</li> <li>- String lowerOne</li> <li>- String lowerTwo</li> <li>- String lowerThree</li> <li>- String upperOne,</li> <li>- String upperTwo</li> <li>- String upperThree</li> <li>- String fileName</li> <li>- String directoryLetter</li> <li>- String fullFileDirectory</li> <li>- String extraDetails</li> <li>- String temp</li>   <li>- intfileHeaderCounter</li> <li>- intfileSize</li> <li>- inttimeLastPlayed</li> <li>- inttimeAdded</li> <li>- intnumberOfTimesPlayed</li> <li>- intcountToFileEntry</li> <li>- inttempCounter</li> </ul>
+ public wmdbExtractorOne(String, int): void

The remaining Class Diagrams are located within *Appendix -Class Diagrams (3.3.8)*

### 3.4.Design of WMP Extractor

Below will discuss the design of the two main functions of WMP Extractor, the extraction of information from the WMP database and the extraction of information of information from the WMP Prefetch files. The design that will be used will be ‘Design a bit, Code a bit, Test a bit’, this has been chosen due to the flexibility that it provides compared to design models such as the ‘Waterfall’ model. By using ‘Design a bit, Code a bit, Test a bit’ it allows for the designing, implementing and then testing of a section before moving on to the next section, whereas the ‘Waterfall’ model requires the design be completed before implementation and testing. Requiring the design to be completed prior to the implementation may cause issues within the implementation or testing sections. If there is an issue during the implementation or testing that cannot be resolved, it may require a complete overhaul of the design to resolve the issue within the implementation. As mentioned above the use of the ‘Design a bit, Code a bit, Test a bit’ model allows for the design to be changed as the project progresses if there are issues within the implementation.

#### 3.4.1.Windows Media Player Database

As discussed within the *Overview of CurrentDatabase\_372.wmdb (2.7)* while media entries have a similar pattern before, after and throughout the WMP database and WMP Prefetch files, they may exist in different locations within the files. This

‘movement’ within the database is dependent on the following known variables, although due to lack of information surrounding both the WMP database and WMP Prefetch as discussed within *Overview of CurrentDatabase\_372.wmdb*(2.7) these may not be the only variables:

- How many times the media is viewed
- When new entries are added to the database
- How the user opens the media files

With no discernable pattern to where entries are written to the WMP database file, it is not possible to design WMP Extractor to extract specific areas within the database as the location of entries could potentially be anywhere. To ensure that no entries are missed within the WMP database *every* byte must be read in and checked to see if it matches patterns described within *Overview of CurrentDatabase\_372.wmdb*(2.7). By checking every byte within the WMP database it ensures that no entries are missed as no part of the database will go unchecked.

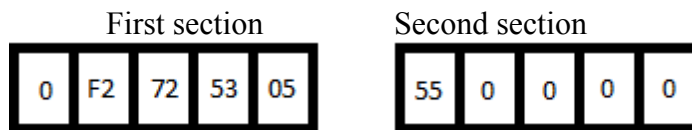
The beginning Hex values for an audio entry are F272 5305 55, several *if* loops will be used to check if the single Hex byte read in at a time matches the first in the pattern. In the above example of an audio entry, WMP Extractor will need to find the value F2. Once WMP Extractor has found the first value in the pattern, it will check the next value to see if it matches the second in the pattern i.e. 72. If the next byte in the process does not match the next in the pattern WMP Extractor will start from the beginning of the pattern. The process will continue until the entire entry pattern has been found. Once WMP Extractor has confirmed an entry it will be able to iterate through the Hex values until it reaches the information to be extracted. An example of this is, once WMP Extractor finds an audio entry it will need to reiterate through 285 bytes of information until it comes to the main audio entry as shown in *Audio Layout* (2.7.1).

Reading in one byte at a time from the WMP database will slow down the analysis process as every byte will be checked compared to skipping to known entries, which is not possible as described within *Overview of CurrentDatabase\_372.wmdb* (2.7). The analysis of the WMP database is slowed down even further by having to check each byte after a byte in the pattern is found. In the case of the audio example if WMP Extractor finds the first four bytes in the pattern but the fifth byte does not match it will have to start to look for the pattern again.

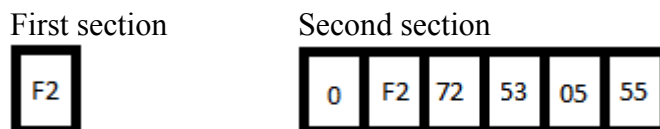
Another way WMP Extractor could perform the analysis of the WMP database is to read in several bytes, store them in memory and check them for the pattern; in the case of the audio entry F272 5305 55. While this solution may seem more efficient by reading in more than one byte at a time, the bytes that would be read in still all need to be checked to see if the pattern exists within them. A problem with this approach is that it may miss entries. If WMP Extractor was to read in five bytes then scan through looking for the pattern F272 5305 55, the first five bytes may only



contain a partial entry with the remaining part of the entry in the next five bytes as shown below.



It would be possible to design `WMP Extractor` to scan through the WMP database until it finds the first Hex value in the pattern; F2. Once `WMP Extractor` has found the first entry from the pattern it would read in four more bytes (or however many more bytes are in the pattern) and check to see if they match the rest of the pattern of that entry; in the case of an audio entry 72 5305 55. The problem with this method is similar to reading in several bytes; `WMP Extractor` may miss the entry. The example below shows that if four bytes were to be read in after the first hex value in the pattern was found it would not match. `WMP Extractor` would then continue scanning from after the four bytes read in and would have missed an entry. Reducing the amount of bytes read in after the first Hex value is found down to three or two would present the same problem.



The only way to mitigate this issue would be to reduce the number of bytes read in after the first hex value is found down to one; this is the suggested first example of how to design `WMP Extractor`. To ensure that there is minimal chance that entries within the WMP database are missed the first design will be used. `WMP Extractor` will read in one byte at a time until it finds the very first byte in the pattern. `WMP Extractor` will then check the next byte to see if it matches the pattern, this process will continue until the entry pattern has been found; thus confirming an entry.

### 3.4.2. Windows Prefetch

As mentioned within *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8), there is no known information available from Microsoft® on how entries are stored within the Windows® Prefetch files. There is one constant with entries that are stored within the WMP Prefetch files; all entries start with `. \ . D . E . V . I . C . E . \ .`

The Windows® Prefetch files have similar problems as the WMP database with the extraction of media entries. Like the WMP database, it appears from research there is no current method *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8) for extracting entries from within Prefetch files that relate to just media entries. Upon investigation undertaken by the author entries within the WMP Prefetch are written to different areas of the file, due to no technical information released from Microsoft® it is not

known which variables affect where entries are written. Entries are different lengths thus WMP Extractor will not be able to extract predefined areas of the file. WMP Extractor must also only extract relevant information from the Prefetch files. This must also be taken into account when designing WMP Extractor.

### 3.5. Human Computer Interface (HCI)

The computer forensic examiner will interact with WMP Extractor through the use of a GUI. Within the scope of WMP Extractor the examiner will be the primary user, as such considerations based on this should be made to provide the best possible interaction between the examiner and WMP Extractor. While the use of Human Computer Interface (HCI) considerations is important when creating the GUI as expressed within *Software Requirements and Justification* (2.11), it is also important that WMP Extractor is a proof of concept and the need for extensive and elaborate GUI is not necessary. The aim is to provide a GUI that is 'intuitive' to use. An intuitive GUI would aim to provide the examiner with a look and feel that mirrors many Windows® applications. By mirroring Windows® application such as the 'File' menu in the top left, it hoped the examiner although never seeing WMP Extractor before will be familiar with similar layout and will have an understanding of how to operate it.

The GUI design for WMP Extractor will conform the User-Centred Design (UCD) model, the standards set out in the UCD require that the GUI focuses on the users (computer forensic examiner) interaction with the software; in the case of this project WMP Extractor (Abrams, Maloney-Krichmar and Preece, 2004). While the GUI design for WMP Extractor will conform to the UCD model, only part conformation is required as WMP Extractor as mentioned above is a proof of concept.

### 3.6. Design of the Graphical User Interface (GUI)

Java allows for the importing of elements from the *Swing Library*, the *Swing Library* is an Application Programming Interface (API) for creating a GUI. The *Swing Library* includes JFrame, JButton, JOptionPane, JInputDialog and JLabel. Using the elements from the *Swing Library* will allow for an effective GUI to be created for WMP Extractor.

The *Swing Library* has been chosen over the Abstract Windows Toolkit (AWT). The AWT library is known to have issue when used across multiple platforms (platforms are different Operating Systems such as Windows®, Mac and Linux). If WMP Extractor used the AWT to create the GUI it may render it unusable if a forensic examiner is using a platform on which it has issues (Liang, Y.D. a 2008). The *Swing Library* is less computer resource intensive compared to the AWT library, this has the advantage of using less Random Access Memory (RAM) therefore minimising the effect if used on a live machine. The second Association of Chief Police Officers (ACPO) guideline allows the forensic examiner to make changes to the original data, as long as they are "able to give evidence explaining the relevance and the implications of their actions" (ACPO, 2010). Minimising the effect WMP

Extractor has on a live machine will require less explanation by the forensic examiner as well as maximising evidential integrity as less data will be altered.

WMP Extractor is being created for the extraction of media viewed upon the Windows® 7 Operating System using WMP. With computer forensic examiners possibly extracting thousands of entries stored within the WMP database and Windows Prefetch, and the possibility of dealing with a high volume of cases, the forensic examiner should have the ability to enter information about the case. Having this ability will help the forensic examiner distinguish one set of search results from another when reviewing over a period of time.

The forensic examiner will be able to enter their name and case number, both of these details will help uniquely identify a search report. Prior to the main GUI being displayed the examiner will be presented with a text box that requires the forensic examiner to enter these details before progressing any further. To create the text box the `InputDialog` from the *Swing Library* will be used. To ensure valid details are entered, characters a-z will only be allowed to be input for the forensic examiners name. Only numbers 0-9 will be allowed for the case number, both boxes will require input from the examiner and won't be able to be submitted blank.

- Examiner name:
  - Will use an `InputDialog` for the input
  - Will error check to ensure field is not blank
  - Will error check to ensure field does not contain numbers
  - Will error check to ensure no special characters are input

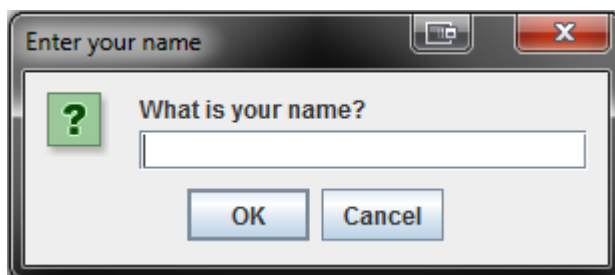


Figure 13- above shows the `InputDialog` the examiner will use to enter their name.

- Case Number/Name:
  - Will use an `InputDialog` for the input
  - Will error check to ensure field is not blank
  - Will error check to ensure field does not contain characters a-z
  - Will error check to ensure no special characters are input

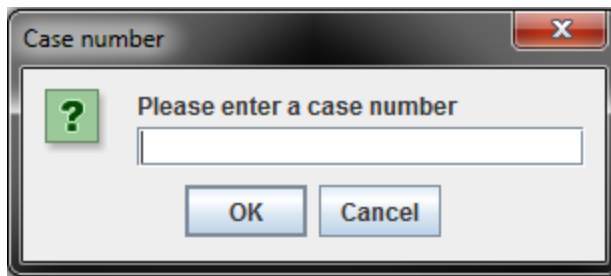


Figure 14 - above shows the InputDialog the examiner will user to enter the case number.

One process that can be automated by WMP Extractor is to record the date and time within the report along with the examiner name, case number and analysis results. To enable WMP Extractor to perform this task automatically, the date and time will need to be taken from the system that WMP Extractor is running on. While this is not an issue if WMP Extractor is being run on a forensic examiners machine as they will be able to set the correct time and date. Using the local date and time taken from a suspects machine when using WMP Extractor as a live tool suffers from disadvantages such as the time being potentially incorrect. While the time may be incorrect when recorded within the report this should be mitigated by the fact that the forensic examiner should be taking contemptuous notes and recording the date, time and what action was taken during the investigation (Anderson P, 2010).

The GUI will only allow the forensic examiner to perform certain actions before being able to progress onto the next stage of an analysis. To create the main GUI of WMP Extractor three main components will be used.

1. JFrame(Main Frame)
2. JMenuBar(Menu Bar)
3. JTextArea(Text Area)

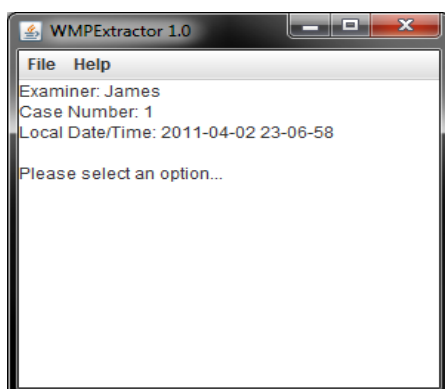


Figure 15 - shows the GUI the examiner is presented with after entering both name and case number.

The menu bar will have two functions: *file* and *help*. The *file* menu will have two options, *extract data* and *exit*. The latter option will allow the examiner to exit WMP

Extractor. The former option *extract data* will presents the examiner with a drop down list with two options. The options will allow the forensic examiner to choose the location of the files to be analysed, the first option *default location* will look for the WMP database and WMP Prefetch files in their default location; as described within *Overview of CurrentDatabase\_372.wmdb* (2.7) and *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8). If the forensic examiner selects the *default location* option WMP Extractor will begin the analysis. The second option *select location* will allow the forensic examiner to choose the folder location be it on the local machine or across a network. Selecting *select location* will present the forensic examiner with a `fileChooser`, a `fileChooser` is similar to the box presented when saving or opening files within Word or other computer program. The use of GUI objects such as `fileChooser` presents the examiner with options they may have encountered before in other software. The *help* menu will provide the examiner with instructions on how to operate WMP Extractor. Providing written instructions the examiner will have more chance of performing the analysis by themselves as the examiner may be able to find the answer to their query.

Error checking within programs is important as it prevents the user from experiencing unexpected scenarios that would reduce the users' experience. While error checking is a background process that the examiner will not be able to see, they should be informed of any errors. An example of error checking built into WMP Extractor is checking to make sure files exist within the selected directory. The file path of both the WMP database and WMP Prefetch will be passed from the `GraphicalUserInterface` class into the `WMDBExtractor` and `PrefetchSearch` class, WMP Extractor will then search for the WMP database and WMP Prefetch using their known naming conventions. The course of action WMP Extractor will take depends on which files are missing from the selected directory. WMP Extractor will be designed to analyse the WMP database first. If the WMP database does not exists within the selected directory the user will be informed of this by the GUI and a note recorded within the report. Once WMP Extractor has established that the WMP database does not exist, it will progress onto the WMP Prefetch files. If WMP Extractor is unable to also find the WMP Prefetch files the forensic examiner will be notified of this via the GUI and a note recorded within report.

During the analysis of the WMP database and WMP Prefetch certain elements of WMP Extractor will be disabled, this will ensure the least as little as possible will prevent WMP Extractor from completing the analysis. If the user is able to select options such on the *file* menu such as, *select location* or *exit* during the analysis of the WMP database and WMP Prefetch it may cause WMP Extractor to fail its analysis.

### 3.7.Design Implementation

This sections will discuss how WMP Extractor was implemented, problems that were encountered during the implementation their solution and what effect these had upon the project. Below describes the implementation of the `PrefetchSearch`

class. The reaming implementation sections can be found within the *Appendix - Design Implementation (5.7)*.

### 3.7.1.PrefetchSearch – class

The purpose of PrefetchSearch is to identify media entries from within the WMP Prefetch files and extract them. Due to the possibility of there being more than one WMP Prefetch file PrefetchSearch was designed to check each file and compare it to the known naming convention of the WMP Prefetch files.

```
for (intfl = 0; fl<listOfFiles.length; fl++){
    if (listOfFiles[fl].isFile()){
        files = listOfFiles[fl].getName();
        if
(files.startsWith("WMP")&&files.endsWith(".pf")){
            }
        }
    }
```

ThePrefetchSearch first gets the number of files within the given folder.

```
for (intfl = 0; fl<listOfFiles.length; fl++)
```

The file names are stored within an array.

```
if (listOfFiles[fl].isFile())
```

The first name is called from the array.

```
files = listOfFiles[fl].getName();
```

The name of the file is then checked against the naming convention of the WMP Prefetch files.

```
if (files.startsWith("WMP")&&files.endsWith(".pf"))
```

If a file name matches the criteria the file is searched for entries. PrefetchSearch looks for the beginning of a file entry \.D.E.V.I.C.E.\. Once a file entry has been found within the WMP Prefetch file, PrefetchSearch looks for either then extension passed to it by PrefetchExtensions thus confirming an entry. If PrefetchSearch finds the end of the file denoted by 0000, before the extension of the file entry being looked for, the search starts again.

An example of how PrefetchSearch would find a MP3 entry within a WMP Prefetch file is show below.

The extension in Hex is passed in from PrefetchExtensionstoPrefetchSearch in this case an MP3 entry.

Extension Letter in ASCII	Extension Letter in Hex
M	4D

P	50
3	33

(Figure 16 – shows the extension of MP3 in both ASCII and Hex)

Once the beginning of a file entry has been found `PrefetchSearch` uses a series of loops to look for the extension passed in by `PrefetchSearch`. By using a series of loops it ensures that every byte within the WMP Prefetch is checked as each byte is read. After the beginning of the file entry the next byte is read and stored within a temporary variable. If the next byte read is the start of the file extension that `PrefetchSearch` is looking for 4D, the next byte is read; this byte is also stored with the temporary variable. Once the beginning an entry has been found `PrefetchExtensions` reads the next byte to see if it matches the start of the file extension e.g. 4D (M in Hex). If the next byte read after the first match, 4D, this is repeated until the extension is found. Once all three Hex values of the extension are found, the hex values stored within the temporary variables are passed to `HexConverter` and the search repeats until the end of the WMP Prefetch file. Once the end of the WMP Prefetch file has been reached, the next WMP Prefetch file is checked. If there are no other WMP Prefetch files to check, the next set of file extensions are received from `PrefetchExtensions` and the process repeats. If any byte read after `PrefetchSearch` finds the beginning of a file entry within the WMP Prefetch matches 00 the next byte is checked, if the next byte after 00 is also 00, the entry that has been found will not have the extension `PrefetchSearch` was looking for.

### 3.8. Testing

Testing is an important step when creating software as it is a measure on how well the software meets its requirements, it is also a measure of its fitness for purpose and the build quality. To enable a measurement of fitness for purpose, the requirements in the analysis are used; fitness for purpose can be measured by asking the question, does the system meet the requirements (Grubb, P. Takang, A.A, 2003). In the case of WMP Extractor the requirements from *Analysis Overview* (2.13) will be reviewed to see if they have been met.

#### 3.8.1. Testing Strategy

Every time a new segment of code was added to WMP Extractor this was tested to ensure it worked as required. Testing WMP Extractor each time a new segment of code was added allowed for the easier debugging of code, if a problem occurred after the new code was added to WMP Extractor that did not exist prior, the problem would usually reside within the new code.

While testing was completed during the implementation of WMP Extractor these tests covered small sections of code, it is still imperative WMP Extractor is tested once complete to ensure the product works as intended.



The type of testing for WMP Extractor will be black box testing. Black box testing is used when the tester has no knowledge of the inner workings of the program such as the source code. WMP Extractor is designed so not only people with computer forensic knowledge are able to use it but for people without this knowledge.

The testing of WMP Extractor should be as close to if it were to be in a real life computer forensic case. By testing WMP Extractor as close to real world conditions will show how it performs and if any problems occur. The closest testing of WMP Extractor to a real life forensics case would be to perform an analysis on both WMP databases and WMP Prefetch files from peoples computers who are running Windows® 7. Due to the ethical considerations this would cause, all test files will be specifically created for this project.

In order to ensure details of entries within the WMP database and WMP Prefetch files were correct, specific files were added to test WMP database and details recorded about the file such as, size, time added and number of views. With specific details known about an entry that resides within the WMP database the details extracted by WMP Extractor should match. The creation of test data over live data also has the advantage of mitigating ethical issues.

### 3.8.2. Windows Media Player Database Testing

To test the accuracy of the results produced by WMP Extractor the WMP database will be used in different states e.g. a blank database or a database with one audio entry. Each time a new test is conducted a new blank database will be used. To ensure the database is blank, a copy of a new database taken from a fresh install of Windows® 7 will be used. Prior to the database being copied over, the MD5 hash value was taken. Each time a copy of this database is used the MD5 hash value is checked against the original to ensure that no changes were made during the copying procedure. The use of a blank database will ensure data from the previous test does not affect the new tests as the only data to reside within the database will be the data added. Before each analysis of the WMP database the MD5 hash value will be noted, once the analysis is complete the MD5 of the database will be compared to the one taken prior to the analysis. In comparing the MD5 before and after it shows if WMP Extractor has made any changes to the database as even if only one byte is changed from a zero to a one the hash value will be completely different.

For each of the tests specific data was noted down prior to adding the media to WMP which subsequently adds an entry into the WMP database. As discussed within *Overview of CurrentDatabase\_372.wmdb* (2.7), each different media type has different information available for extraction. Details of the test files that were recorded prior to them being added to WMP, once the media files were added to WMP an entry will be created within the WMP database. WMP Extractor was then used to analyse the WMP database and the details returned checked against the details of the test file. The MD5 hash value was taken after the media file was added to WMP and then again after the analysis of the WMP database by WMP Extractor. *Figure 17* below shows an audio file that was added to the WMP database, the file information recorded before and the file information returned by WMP Extractor.

Whilst the time taken for media files to be added and played included seconds, these have been omitted from the test results as there is a delay from when the file was added to the WMP and when the entry is recorded within the WMP database.

Type of Information	File Information Entered	File Information Returned
File size in bytes	8,414,449	8,414,449
Date and time added to WMP	29/02/2011 16:04	29/02/2011 16:04
Date and time audio file was last played	29/02/2011 16:04	29/02/2011 16:04
Number of views	0	0
File name	Kalimba.mp3	Kalimba.mp3
Full file path and name of file	C:\Users\James\Pictures\Sample Music\ Kalimba.mp3	C:\Users\James\Pictures\Sample Music\ Kalimba.mp3
MD5 Hash	F3504348A65E6B47B686A4C2FFB8843D	F3504348A65E6B47B686A4C2FFB8843D

(Figure 17)

The remaining testing of *WMP Extractor* analysis of the WMP database can be found within the *Appendix - Windows Media Player Database Testing* (5.8.1).

### 3.8.3.Windows Media Player Prefetch Testing

Below will discuss the testing of *WMP Extractor* when, performing an analysis upon the WMP Prefetch files. As mentioned within *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8) it is not possible to predict when WMP will add entries to the WMP Prefetch as there is currently no information available from Microsoft®. Without the ability to add entries to the Prefetch files, it would be not be possible to test that *WMP Extractor* is able to recover every different type of file from them. To enable the ability to test *WMP Extractor* using all different file type entries within the Prefetch will be modified to resemble entries as if they were created by WMP using a hex editor. The entry below is an example of an audio entry within a WMP Prefetch file.

```
\D.E.V.I.C.E.\H.A.R.D.D.I.S.K.V.O.L.U.M.E.4.\U.S.E.R.S\J.A.M.E.S.\M.U.S.I.C.\T.E.S.T.M.P.3\
```

To create test entries for all different media types the extension of a file was changed using a Hex editor. Below shows an example of the audio entry above, that has been changed to an image entry. While the test data that has been changed is synthetic compared to naturally being placed there by WMP, it is none the less the same allowing for the testing of all file types.

```
\D.E.V.I.C.E.\H.A.R.D.D.I.S.K.V.O.L.U.M.E.4.\U.S.E.R.S\J.A.M.E.S.\M.U.S.I.C.\T.E.S.T.P.N.G\
```

Compared to the testing of the WMP database *Windows Media Player Database Testing* (3.8.2) *WMP Extractor* only extracts the full file path for entries from within the Prefetch files. The test for the WMP Prefetch files must ensure that *WMP*

Extractor extracts the full file path. WMP Extractor is designed to search for multiple WMP Prefetch files due to the possibility of there being more than one upon a suspect's machine as shown within *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8). Each file type had an entry created within a WMP Prefetch of its own to also allow the testing of the possibility of multiple WMP Prefetch files. As with the testing of the WMP database *Windows Media Player Database Testing* (3.8.2) the MD5 hash value of the WMP Prefetch file was taken prior to and after the analysis to ensure to modification to the file had taken place.

#### 1. PNG

Data within Prefetch file	Data extracted by WMP Extractor
\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.P.N.G\	\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.P.N.G\
MD5 Hash value before analysis	MD5 Hash value after analysis
67F48C64AC48EE5D63742F8F2C5B32 B7	67F48C64AC48EE5D63742F8F2C5B32 B7

All tests conducted show that WMP Extractor is able to extract entries from within the WMP Prefetch files without altering the information before being output into the report. The tests also show that WMP Extractor does not alter the WMP Prefetch files. Both of these test show WMP Extractor maintains evidential integrity during the extraction of information from the WMP Prefetch files. The searching of multiple WMP Prefetch files confirms that if a WMP Extractor was to be used within a forensic case, it would be able to handle the possibility of multiple WMP Prefetch files.

The remaining test can be found within *Windows Media Player Prefetch Testing* (5.8.2).

#### 3.8.4. Test Cases

The Use Cases' in *Use Case Description* (3.3.6) will be used to create multiple test cases' for WMP Extractor. Using the Use Cases as well as other criteria will ensure a full range of test cases' are created. Using extensive testing ensures that all requirements for WMP Extractor are tested, by testing every function of WMP Extractor it ensures that it is fit for purpose. The testing of WMP Extractor will show that it either performs as expected by producing correct results or not. If WMP Extractor does produce correct results within the testing stage results produced when used in a real world situation can be trusted.

Figure 18 below is a Test Case showing the testing of the examiner name box that appears when WMP Extractor is run. By using Test Cases it is clear to see if WMP Extractor passed the test or not by comparing the expected result to the actual result. The recording of the actual result helps to problem solve any errors with WMP

Extractor by describing what happened and comparing it with the expected result. The below Test Case shows the test of leaving the examiner name blank, the expected result matches the actual result confirming the test was successful.

Unique ID	Test 01
Test Description	Examiner name cannot be left blank
Pre-Conditions	Programming running
Test Data Used	Examiner name filed left blank
Expected Result	User advised examiner filed was blank
Actual Result	User advised examiner name was blank

(Figure 18 – a test case showing the build quality of WMP Extractor with regards to error handling)

### 3.8.5. User Testing

To gain an understanding of how ‘intuitive’ WMP Extractor is to use in the sense of little to no training user feedback was sourced. WMP Extractor is a proof of concept and while results gained from user testing will highlight any areas for improvement. Five computer forensic students were given a USB memory stick with WMP Extractor on, and a laptop with both the WMP database and WMP Prefetch files in their default locations as discussed within *Overview of CurrentDatabase\_372.wmdb* (2.7) and *Overview of WMPLAYER.EXE-xxxxxxx.pf* (2.8). All five computer forensic students were asked to perform both a simulated live analysis where they would extract information about media viewed using WMP from the default locations, and a static analysis where a folder containing the same files was present upon the desktop.

Once each student had completed both a simulated live and static analysis of both the WMP database and WMP Prefetch files, they were asked for their views on WMP Extractor. User testing revealed that all five computer forensic students were able to operate WMP Extractor with no training, although this may not be a conclusive test as all students have an advanced level of computer knowledge. It was mentioned by one of the computer forensic students that while the ability to not use the file menu during the analysis seems like a good idea, “it gave the impression that the program had crashed”. The student mentioned that when other applications have crashed while in use the program becomes inaccessible; including the menu, the student advised some sort of feedback would resolve this issue. Another computer forensic student mentioned that on their first analysis of the WMP database and WMP Prefetch files, it was unclear of the time the entire analysis would take. This student said they were unsure if they should start another task during the analysis or if they had time “to make a cup of tea”. The computer forensic student suggested that a ‘status bar’ be implemented, even if it were approximate to resolve this issue. By informing the examiner of how long an analysis will take they will be able to manage their time better which will go some way towards reducing their backlog.

It should be noted that it was mentioned in passing by two of the computer forensic students that, they had not seen any software that provides this functionality and that it should be developed further including other versions of WMP. All tests conducted during user testing completed successfully, while the tests were designed to receive feedback into the ‘ease of use’ of the GUI of WMP Extractor as laid out within *Software Requirements and Justification* (2.11), it reinforces the effectiveness of WMP Extractor as shown within the *Testing* (3.8).

## **4. Evaluation**

The evaluation section will be broken down into; the evaluation of WMP Extractor, the evaluation of the project process and the evaluation of testing. The evaluation of WMP Extractor will discuss the strengths and weaknesses of WMP Extractor both build quality and fitness for purpose. The evaluation of the project process will discuss what was successful and not successful during the project process. This section will also cover the project plan vs. the route taken as well as the personal progress of the author.

#### 4.1. Evaluation of Fitness for Purpose

This section will use the requirements from *Analysis Overview* (2.13) and compare them to the outcomes to analyse WMP Extractor fitness for purpose. As discussed within the *Analysis Overview* (2.13) the requirements for this project are.

1. Extract information about media viewed upon Windows® 7
  - a. Extract information from the *CurrentDatabase\_372.wmdb*
  - b. Extract information from *WMPLAYER.EXE-xxxxxxx.pf* files
2. Ensure the evidential integrity is maintained when extracting information
  - a. Extract exact copies of entries from the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
  - b. Ensure no alterations are made to either the *CurrentDatabase\_372.wmdb* and *WMPLAYER.EXE-xxxxxxx.pf*
3. Produce a report for easy analysis
4. Have Graphical User Interface (GUI) to allow for ease of use

Below will discuss each requirement individually, and judge if the requirement was completed. The outcomes will be compared to the requirements to see how successful they were by seeing if the outcome matches the requirement.

1. Extract information about media viewed upon Windows® 7

The extraction of information about media viewed upon Windows® 7 has two parts to it. WMP Extractor was required to extract information from the Windows Media Player (WMP) database *CurrentDatabase\_372.wmdb*. It was also required to extract information from the Windows® Prefetch files *WMPLAYER.EXE-xxxxxxx.pf*.

- a. Extract information from the *CurrentDatabase\_372.wmdb*



The first requirement of WMP Extractor was to extract information about media viewed using WMP. The first part is the extraction of information from the WMP database about media viewed using WMP. To judge if the extraction of information from the WMP database was successful, the information found to be contained within the database in *Overview of CurrentDatabase\_372.wmdb* will be compared to the data that WMP Extractor is able to extract.

There are three different media types stored within the WMP database; audio, image and video. As described within *Overview of CurrentDatabase\_372.wmdb*(2.7), each different type of media has different information stored about it within the WMP database. For WMP Extractor to be successful in the extraction of information about media viewed within the WMP database it must extract all the information from each media type presented within the *Overview of CurrentDatabase\_372.wmdb* (2.7).

- Audio Entries

The information that is available from the audio entries within the WMP database as shown in *AudioLayout* (2.7.1) are:

- File size in bytes (*not* the size on disk)
- Time media file added to WMP
- Time media file last viewed
- Number of times media file viewed
- File name
- Full media file name and path
- Extra details about media file

*Figure 19* within the *AppendixImages* (5.3) shows the extracted data in the report created by WMP Extractor for several audio entries. The extracted data within the report includes all of the above information. With all data known that could be extracted; extracted from the WMP database for an audio entry, it is safe to conclude that WMP Extractor passes the requirement of extracting audio entries from the WMP database.

- Image Entries

The information that is available from the image entries within the WMP database as shown in *Image Layout* (2.7.1).

- File size in bytes (*not* the size on disk)
- The year the file was created

- The width of the file in pixels
- Time media file added to WMP
- File name
- Full media file name and path
- Extra details about media file

*Figure 20* within the *AppendixImages* (5.3) shows the extracted data in the report created by *WMP Extractor* for several image entries. The extracted data within the report includes all but two of the information available for extraction, the year the photo was taken and width of file in pixels.

As discussed within *Image Layout* (2.7.2) the year the file was created and the size in pixels will not be extracted from the WMP database. These as discussed are not being extracted as they are information about the file not information about how/when the file was viewed using WMP. While the same could be said about the size of the file, this size has been included for extraction as it will help in the aid of recovering the file from unallocated cluster (where deleted files reside). If the forensic examiner is able to find the file in unallocated clusters, by using the known file header for an image file; (e.g. yoya for a .jpg image file) they will be able to work out how many bytes need to be extracted from file header as this is always in a predefined position within the image file.

With all the possible data within the report being extracted that could be extracted by *WMP Extractor*, excluding the agreed two data fields, it is safe to conclude that *WMP Extractor* passes the requirement of extracting image entries from the WMP database.

- Video Entries

The information that is available from the video entries within the WMP database as shown in *VideoLayout*(2.7.3).

- File size in bytes (*not* the size on disk)
- Time media file added to WMP
- Number of times media file viewed
- File name
- Full media file name and path
- Extra details about media file

*Figure 21* within the *AppendixImages* (5.3) shows the extracted data in the report created by WMP Extractor for several video entries. The extracted data within the report includes all of the above information. With all data known that could be extracted; extracted from the WMP database for an video entry, it is safe to conclude that WMP Extractor passes the requirement of extracting video entries from the WMP database.

The above tests show that WMP Extractor is sufficient in extracting all data fields for audio, image and video. What these test do not show is that the information is correct. To show that the data extracted by WMP Extractor is correct several tests were conducted within *Windows Media Player Database Testing* (3.8.3). The tests conducted were designed to ensure that the data that WMP Extractor; extracts from the WMP database is identical that that was stored within the WMP database. Each test shows that data that inserted into the WMP database by adding a file to WMP, each test also shows the data that was extracted by WMP. All media types including audio, image and video were tested. All tests conducted extracted identical information to what was in the WMP database. With WMP Extractor extracting information that is identical to what is stored within the WMP database, it can be concluded that evidential integrity is maintained by WMP Extractor.

By WMP Extractor being able to extract all three media types; audio, image and video, not only in ensuring that all data fields are extracted, but the results produced shown to be accurate, WMP Extractor passes the requirement of being able to extract information about media viewed using WMP.

The second part of the first requirement was the extraction of information about media viewed using WMP stored within the WMP Prefetch files.

a. Extract information from Prefetch files

To judge if the extraction of information from the WMP Prefetch files was successful, the information found to be contained within the database in *Overview of WMPALYER.EXE-xxxxxxx.pf*(2.8) will be compared to the data that WMP Extractor is able to extract.

As discussed within *Overview of WMPALYER.EXE-xxxxxxx.pf* (2.8) it is not known completely under what conditions entries are added to the WMP Prefetch files due there being no current information released from Microsoft®. While it appears to have been discovered that entries are added to the WMP Prefetch files when they are played without importing them as described within *Overview of WMPALYER.EXE-xxxxxxx.pf* (2.8), this is not always true. With the difficulty of creating test data asdiscovered within *Windows Media Player Prefetch Testing* (3.8.3) it was not possible to test WMP Extractor using every different media type that was created by the use of WMP with the Prefetch files. As discussed within *Windows Media Player Prefetch Testing* (3.8.3) due to the lack of test data for the WMP Prefetch, a WMP Prefetch file was edited using a Hex editor. The creation of WMP Prefetch test data should not be a reason for failing the second part of the first requirement because

it has not been tested using real world test data. The test data created by changing the entries within the WMP Prefetch would be the same as if the entries were placed there themselves by WMP themselves.

WMP Extractor is able to extract entries about media viewed from the WMP Prefetch files for all the different types of media tested including .png, .jpg, .gif, .bmp, mp3, .aac, .wav, .mpg and .avi. WMP Extractor is also able to extract information across multiple WMP Prefetch files about media viewed using WMP, this will ensure that no entries are missed. With WMP Extractor being able to successfully extract information about media viewed using WMP from the WMP Prefetch file; it demonstrates that it passes the second part of the first requirement.

WMP Extractor is able to display that it meets the first requirement *Extract information about media viewed upon Windows 7* by showing that it is able to complete both the extraction of information from the WMP database and WMP Prefetch files accurately.

## 2. Ensure the evidential integrity is maintained when extracting information

Evidential integrity is the importance of ensuring that original evidence is not changed in any way; either by deliberate or non-deliberate actions. Evidential integrity must be maintained throughout the analysis process to ensure that results produced by the analysis are admissible in court. If evidential integrity is seen to have been compromised in anyway, results will be inadmissible in court as they will be seen to be inaccurate.

Evidential integrity was conducted during the testing within *Windows Media Player Database Testing* (5.8.1) and *Windows Media Player Prefetch Testing* (5.8.2) of WMP Extractor. Evidential integrity was tested in two ways.

The first was using the MD5 hash algorithm. By taking the hash value prior to the analysis of both the WMP database and WMP Prefetch files and after the analysis, it was possible to compare any changes made to the files as the hash values would have been different. WMP Extractor is successfully able to perform an analysis upon both the WMP database and WMP Prefetch files without modifying them. The second way that the evidential integrity of WMP Extractor was tested was tested within the *Testing* (3.8) was, to compare the test data created for the both WMP database and WMP Prefetch against the results produced. During the testing procedure WMP Extractor produced results that were identical to the test data created for both the WMP database and the WMP Prefetch files.

WMP Extractor successfully fills the requirement *Ensure the evidential integrity is maintained when extracting information* by not changing both the WMP database and WMP Prefetch files during the analysis, as well as producing accurate results as shown within *Windows Media Player Database Testing* (3.8.2) and *Windows Media Player Prefetch Testing* (3.8.3).

## 3. Produce a report for easy analysis

WMP Extractor produces a report in a Tabbed Separated Value (TSV) file that can be read by Excel as discussed within *DesignImplementation* (3.7). Information extracted from the WMP database is written to the report with each record set along a single row, different values of information are written using several columns as shown in *Figures 19, 20 and 21* within the *Appendix -Images* (5.3). Books, websites and newspapers are written and read from left to right within the United Kingdom and other Western countries. By providing a report that follows these rules as well as separate data into rows and columns, it will create a report that is easy to read due to data being distinguishable from other data.

To test this same five students used within *User Testing* (3.8.5) were asked to review a report produced by the author that contained data extracted from the WMP database and WMP Prefetch files. The five students were asked if the report was 'easy' to read. While the general consensus was yes; the main criticism was that while the data was separated within different cells, it would be better if a border was used around the data. Overall the report produced by WMP Extractor fulfilled its requirements to provide the examiner with a report that is easy to read.

#### 4. A Graphical User Interface (GUI) to allow for ease of use

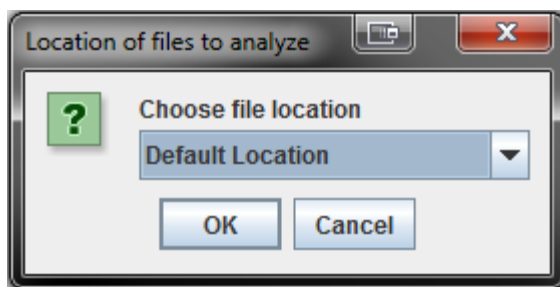
As described within the *Design of the GraphicalUser Interface* (3.6) the GUI was been designed to be as simple as possible by providing the forensic examiner with least amount of options to select. Options within the GUI are also disabled depending on what stage the analysis is preventing mistakes from being made. The ease of use was further increased by the on screen instructions and help section within WMP Extractor. In order to gauge the effectiveness of WMP Extractors GUI, five forensic students were asked to perform a simulated live and static analysis of the WMP database and WMP Prefetch files. Results from these tests can be found within *User Testing* (3.8.5). Overall the feedback from the users suggested that the GUI was sufficient but suggested areas that could be improved upon.

#### 4.2. Evaluation of Build Quality

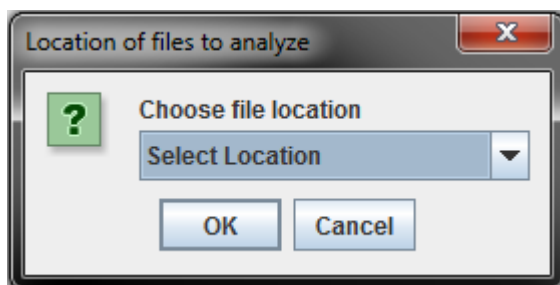
This section will use the test cases in *Test Cases* (3.8.4) to determine the build quality of WMP Extractor. By using the test cases it is possible to determine the build quality of WMP Extractor by evaluating which test cases were successful and which were not. Test cases that were successful will help determine WMP Extractors strengths and un-successful test case will help determine its weaknesses. Test cases carried out were conducted while performing an analysis of both the WMP database and WMP Prefetch files; this allowed the test cases to take the flow of a forensic examiner using WMP Extractor. As discussed above the build quality of WMP Extractor will be based upon the test cases carried out, the evaluation will also follow the flow of the test cases show in *Test Cases* (3.8.4).

Test cases one to eight test WMP Extractors validation checking for both the examiners name and case number. The tests conducted checked that if an examiner entered invalid data such as numbers within their name then the input was rejected. Not only was invalid data tested but valid data was also tested. In all eight test cases WMP Extractor performed expected.

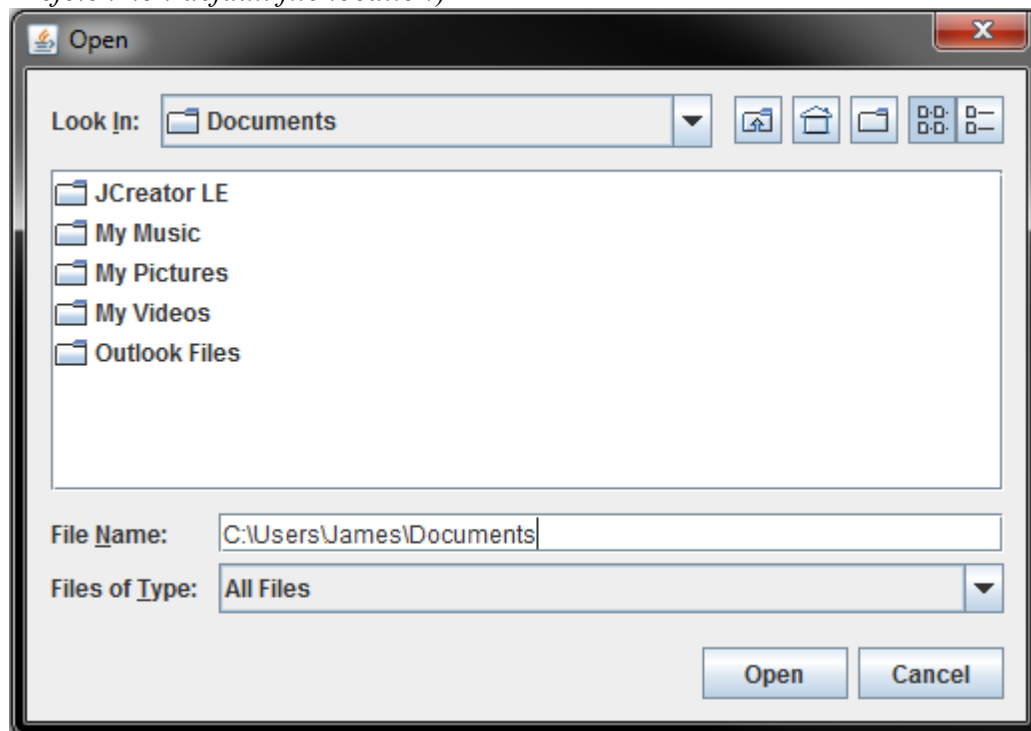
Test cases nine and ten tested that the examiner is able to select both the “Default Location” and “Select Location” option in the GUI. An analysis of the WMP database and WMP Prefetch files was conducted; the analysis was conducted using both the default file location and a different file location was selected. In both tests WMP Extractor was able to find the files located within each directory used. *Figure 22, 23 and 24* below show the options available to the examiner when selecting either the default file directory or selecting the directory.



(Figure 22 – shows the examiners ability to use the WMP database and WMP Prefetch default file location)



(Figure23 – shows the examiners ability to use the WMP database and WMP Prefetch non-default file location)



(Figure 24 – shows the File Dialog box presented when the examiner selects “Select Location”)

Test cases eleven to fifteen tested WMP Extractors ability to handle non present files. The test was designed to ensure that if either the WMP database, WMP Prefetch or both files were missing from the selected directory WMP Extractor was able to handle this and not crash. All five tests showed that WMP Extractor is capable of handling missing files.

Test cases sixteen to seventeen tested the ability to use the *Exit* and *Help* options from the *File* menu. Both tests performed as expected and can be considered successful.

Test cases above have been designed to test WMP Extractors build quality. As mentioned above the build quality of WMP Extractor would be based on which test cases passed and which failed. As shown above all test cases passed, with no test cases failing it is safe to conclude that WMP Extractor meets the build quality designed.

#### 4.3. Evaluation of Project Process

This section will discuss the success of the project overall. Topics that will be covered are the project plan as laid out in the *Terms of Reference* (5.1) within the *Appendix(5)*, comparison to the route taken during the undertaking of the project. The project plan vs. the route taken will discuss the impact of decisions made during the project and their impacts. The personal progress of the author will also be also be discussed and include how the authors knowledge base has increased and what changes the author would make to the project if it were to be completed again.



#### 4.3.1.1. Project Plan vs. Route Taken

Using the recommended time frame of four hundred hours to create this project, a time plan was drawn up as shown within the *Appendix Terms of Reference* (5.1). The time plan drawn up used the total four hundred hours and divided them up between specific tasks. Whilst a best guess was made on the estimation of completion for each task, it was difficult to judge each tasks time frame due to a project of this scale never being undertaken by the author before. Tasks such as the research into the layout of WMP database took considerably longer due to the lack of previous research available and the proprietary format the data is stored in. The estimation of each task was further hampered by the scheduling of other modules within the degree being undertaken by the author.

With the project plan providing an estimated guide for each project milestone, each milestone has either been met within its time frame or exceeded slightly. While a milestones time frame may have been exceeded, there has been no discernable effect upon the project as a whole as time was made up within other areas. With milestones being met the project was completed within the given time frame. While the project has been completed within the specified time frame and time allocated to the evaluation of the project, an increased time frame after the project was completed would have been a considerable advantage. An increased time frame once the project was completed would allow for new features within WMP Extractor to be implemented to increase its functionality as shown within *Further Work* (4.2), as mentioned above this was not possible due to other commitments by the author.

The project as a whole has shown the need for the continued professional development not only by the author but the forensic community as a whole (the authors personal progress is discussed within *Personal Progress* (section)). As noted within (section), there is currently no known solution to the extraction of information about media viewed using WMP within Windows® 7. Windows® 7 has been out in either Beta, RC1 or for purchase since January 2009, with what appears to be the first consideration towards a solution the forensic community should ensure they stay 'ahead of the game' and *not* be required to play catch up.

After the large complex process of research into the WMP database and WMP Prefetch file, the design and implementation section was the next time consuming process. The authors previous knowledge with Java helped reduce the implementation time of WMP Extractor as the syntax of Java was already known. While some areas of the construction of WMP Extractor were complex due to the proprietary format data was stored in as described within *Design Implementation* (3.7), the author's previous knowledge within Java enabled for a better understanding when dealing with issues outside the scope of the author's knowledge.

Once WMP Extractor had been constructed a suitable testing strategy was devised as shown within *Testing* (3.8). Prior to undertaking *Computer Forensics* at Northumbria University the author spent several years within industry which included time testing software, although none of it was written by the author it provided a solid starting ground. The testing of WMP Extractor throughout the development stage



enabled any problems to be removed prior to its completion. Once WMP Extractor was complete it was tested rigorously, while the testing during the development may have slowed down its implementation, it enabled time to be saved during testing as no problems were found that needed to be corrected. While the testing plan was not as strictly adhered to throughout the entire project, it helped organise the workload for this project as well as other modules to ensure that the best use of time and maximize productivity.

#### 4.3.1.2. Personal Progress

While the author has undertaken a significant amount of research during their time at University, the scale and complexity of this project has provided an excellent learning experience that has widened the author's knowledge considerably. The continued personal development within the computer forensic industry is important to ensure that examiners stay current with new technologies. Staying current with new technologies helps increase the examiners flexibility in dealing with more current technologies. The continued personal development should be fulfilled through both the use of training and research. As mentioned above within *Project Plan vs. Route Taken* (4.1.3.1), the research into the extraction of media viewed using WMP within Windows® 7 seems to have been overlooked by the computer forensic community. It is hoped that the research conducted by the author will go some way towards helping the computer forensic community play catch up, as well as build upon the research conducted throughout this project.

The author's skills have increased not only within computer forensics but within a general computing sense. As with most tasks, the practicing and use of skills will improve them. As discussed within *Identification of Tools and Techniques* (2.12) WMP Extractor was written using Java. Prior to undertaking this project the author was taught Java in both the first year in *Programming One* (CG0047) and *Programming Two* (CG0048), and in the second year *Data Structures and Algorithms* (CM055) at University. The author's knowledge and capability of Java has been greatly expanded throughout the project process, this relates to the different classes that are available in Java for use in solving a given problem. Prior to the undertaking of this project the knowledge of the need to develop software was met with reluctance. During the development stage of WMP Extractor the author's knowledge within Java increased, this increase led to a more interested outlook on the project and in turn increased productivity. As well as programming the author now has a greater understanding of the technical aspects of Windows® 7. Knowledge gained about Windows® 7 over the course of this project includes how data is stored within Windows® 7, , in addition to Windows® Prefetch files work as a whole not just WMP Prefetch files and their importance in an investigation.

The design process was not seen as of a great importance prior to the undertaking of this project. During the design process the relevance of a good design was still not clear to the author. Once the implementation of WMP Extractor started the authors view on its relevance shifted to understanding its purpose in the life cycle of software development. It is believed that the use of a design within this project has helped reduce the amount of time during the implementation of WMP Extractor.

The author's research skills have greatly increased throughout this project. Prior to this project the author had undertaken research into writing reports and problem solving both at University and time within industry. Research into both of these areas has been provisionally where material already exists on the given subject, this project enabled an insight into conducting what appears to be original research as discussed within *Initial Research* (2.3). With research into the WMP database file taking longer than expected, the management of time was of grave importance to ensure that all aspects of the project were completed. Skills both learnt and improved upon throughout this project will benefit the author throughout their career; the skills will also enable the continued professional development that is vital in ensuring that the author stays current within the computer forensic community.

#### 4.3.2. Evaluation of Tools and Techniques

Within *Identification of Tools and Techniques* (2.12) the tools needed to create WMP Extractor and the techniques used in its construction were chosen. This section will evaluate the tools and techniques chosen and suggest possible approaches that could have been used in the construction of WMP Extractor.

Java was selected as the programming language to construct WMP Extractor in due to it being an Object Orientated language, author's familiarity with it and its ability to read in files using the built in Hex class. The use of Java throughout this project has been in large part a success, there have been no issues with regards to Java not being able to perform a given task. Java's ability to read in files using the Hex class has enabled a simpler implementation of both the WMPDBExtractor and PrefetchSearchClasses' as described within *Design Implementation* (3.7). While Java has performed well in the construction of WMP Extractor, its need for the Java Virtual Machine (JVM) to be installed could cause issues with WMP Extractor being used upon a live machine. If the JVM is not installed upon a machine that requires a live analysis WMP Extractor will not be able to run. As mentioned within *Possible Approaches* (2.10), it is possible to perform an analysis of both the WMP database and WMP Prefetch using the 'static' method. Performing a static analysis a considerably longer time to perform, and the ability not to be able to perform a live analysis without the JVM reduces the flexibility of WMP Extractor to be used in different situations. WMP Extractor has been designed for the extraction of information about media viewed using WMP within Windows® 7, all versions of Windows® 7 come with the Microsoft .NET Framework 3.5 installed (Microsoft® Corporation, j 2011).

To overcome the need for the JVM when using programs such as WMP Extractor written in Java, Java could be substituted for C#. The advantage of C# over Java is that it does not require the JVM and only requires the .NET Framework that comes installed upon Windows® 7. By using C# it would ensure that a live analysis is hampered by the need for third party software. If the programming language in which WMP Extractor was constructed changed to that of the example mentioned above C#; the design language and design model used would need to be reassessed. This project used UML as the design language and the 'Design a bit, code

a bit, Test a bit' model. UML was chosen as the design language as it is able to represent Object Oriented programming languages such as Java used within this project. As mentioned above if C# were to be used in the construction of WMP Extractor, due to C# being an Object Oriented programming language, as well as being derived from the C programming language like Java; the use of UML as a design language would still be permissible within the scope of this project. While the UML diagrams used within this project have been designed for use with an Object Oriented programming language, they have *not* been designed specifically for use with Java. With this in mind the current UML diagrams could be used if WMP Extractor were to be re-written using C#. As mentioned above the design model used within this project was the 'Design a bit, code a bit, Test a bit' model, while the 'Waterfall' model was considered it was deemed not suitable for this project due to its inflexibility. If C# were to be used in the construction of WMP Extractor both the 'Design a bit, code a bit, Test a bit' and 'Waterfall' model could be used. As with using the 'Design a bit, code a bit, Test a bit' model with Java it allows for a more flexible approach as the design can be altered throughout the construction of WMP Extractor as needed. Originally the 'Design a bit, code a bit, Test a bit' model was chosen for use in this project as it was not clear if the design would work straight away, knowing that the design of WMP Extractor did work, and that C# is from the same family language as Java (C) the use of the 'Waterfall' would almost certainly work.

To describe the interaction between the examiner and WMP Extractor a use case diagram was used as shown within *Use Case Diagram* (5.6). The use case diagram used one use case description to show the extraction of media entries from the WMP database, one use case description to use to describe the extraction of media entries from the WMP Prefetch files. As shown within *Design Implementation* (3.7) WMP Extractor was implemented so that for the extraction of information from the WMP database and WMP Prefetch files, one *class* would supply the details required by another *class* to extract the media entries from both files. While it would have been possible to design the use case diagram to take this into account, due to only one *class* for the extraction of information from the WMP database and WMP Prefetch files it was deemed appropriate to use one use case description.

The Independent Development Environment (IDE) of choice for this project has been JCreator (JCreator2010). As described within *Identification of Tools and Techniques*(2.12) JCreator was chosen due to the author's familiarity and the free version available for academic use. While JCreator is only able to handle the Java programming language only unlike other IDE's, it performed flawlessly throughout the project and only crashed when the author made logical errors in programming such as infinite loops. If a different language were to be used as discussed above the use of JCreator will need to re-evaluated and another IDE sourced for use such as Microsoft's® Visual Studiowhen using the C# programming language (Microsoft® Corporation, k 2011).

#### 4.3.3.Evaluation of Testing

Below will discuss the testing carried out within *Testing* (3.8) and evaluate how successful it was, below will also discuss alternatives to the testing already used and the benefits compared to the testing used.

Two types of testing were used within this project to test WMP Extractor. The first set of tests were used to evaluate WMP Extractors fitness for purpose, these tests used test data to ensure the results produced from analysis of both the WMP database and WMP Prefetch were accurate as shown in *Windows Media Player Database Testing* (3.8.2) and *Windows Media Player Prefetch Testing* (3.8.3). The second set of tests were conducted to test the build quality of WMP Extractor. The build quality of WMP Extractor was conducted by testing the outcomes out of operations from WMP Extractor to the expected outcomes as show within *Test Cases* (3.8.4).

The testing of WMP Extractor of both its fitness for purpose and build quality showed that while WMP Extractor is a proof of concept it would be able to perform an analysis upon a real forensic case. While the results that WMP Extractor produces are accurate, results may not be admissible in court due to the way the testing was carried out. The author has no prior experience with testing software. With no previous experience of testing software it may be argued that the author is not competent enough to perform the testing. If the author is shown not to be competent enough to perform software testing, results may be rejected as the testing of WMP Extractor may be wrong thus the results it produces may be wrong.

A solution to allow for a more professional testing to be carried out on WMP Extractor would be, to submit it for testing via the National Institute of Standards and Technology (NIST) Computer Forensic Tool Testing (CFTT) (Computer Forensic Tool Testing, a 2003) or the Software Testing Standard SIO/IEC 29119 by the British Standards Institute (BSI) (British Standards Institute, a 2011).

The CFTT section of NIST conducts testing on both hardware and software. NIST are a government run organisation with the purpose of developing and applying technology, measurements and standards. NIST were founded in 1901 and have made contributions to projects including DNA diagnostic, smoke detectors and atomic clocks (National Institute of Standards and Technology 2011). If WMP Extractor were to be submitted to the CFTT section of NIST it would provide credible evidence of the testing carried out. NIST have a proven track record within the area of testing critical technologies. NIST conduct extensive testing of both hard and software, there testing procedure is documented below.

1. NIST acquires the tool to be tested.
2. NIST reviews the tool documentation.
3. NIST selects relevant test cases depending on features supported by the tool.
4. NIST develops test strategy.

5. NIST executes tests
6. NIST produces test report.
7. Steering Committee reviews test report.
8. Vendor reviews test report.
9. NIST posts support software to web.
10. NIJ posts test report to web

(Computer Forensic Tool Testing, b 2003)

If `WMP Extractor` were to pass the NIST testing, with their credible and long history within testing software; doubts over `WMP Extractor` testing procedure would be quashed.

`WMP Extractor` could also be submitted to the BSI. The BSI states it “offers a full range of testing and certification services from research and development, pre-production, prototype, batch and audit to full compliance testing” (British Standards Institute, b 2011). The BSI was also founded in 1901; they cover Europe, the Middle East, Africa, Asia and America covering 147 countries. The BSI provides the *Kitemark* certification as well as CE compliance within the EU (British Standards Institute, c 2011). If `WMP Extractor` were to be submitted to the BSI it would as with the NIST provide credible evidence of comprehensive testing. With the BSI proving testing and accreditation to life saving equipment such as healthcare, medical devices and personal protective equipment standards will be of the highest nature (British Standards Institute, d 2011). The BSI creates and implements the standards that other companies follow. By `WMP Extractor` being tested by the BSI any doubts over the quality of testing would be quashed as the BSI’s business model is built on quality and standards.

Due to the time frame of this project and the possible costs involved `WMP Extractor` has not been submitted to the NIST or BSI. Testing offered by both the NIST and BSI offer superior quality testing compared to that undertaken within this project, the testing undertaken within this project has been deemed fit by both the author and project supervisor. If `WMP Extractor` were to be developed beyond a proof of concept, both testing via the NIST or BIS are viable routes of testing.

#### 4.4. Further Work

While `WMP Extractor` is a proof of concept, it is able to successfully extract information about media viewed using Windows Media Player (WMP) upon Windows® 7. `WMP Extractor` is successful in terms of meeting the requirements as set out in *Analysis Overview* (2.13) and shown within *Evaluation*(4). Future work into `WMP Extractor` only serves to increase its usability by implementing greater functionality and its use in alternative scenarios. Below will discuss the further work that could be carried out into `WMP Extractor` including improvements and extra



features as well as any further work into research about media viewed upon Windows® 7. This section will also evaluate alternative uses for WMP Extractor and possible approaches to them.

#### 4.4.1. Alternative uses for WMP Extractor

Below will discuss alternative uses for WMP Extractor. While primarily designed to extract information about media viewed using WMP upon Windows® 7, WMP Extractor may be able to perform other tasks extending its usability outside just forensic software.

WMP Extractor is able to extract media entries from within multiple WMP Prefetch files, as discussed within *Windows Media Player Prefetch Testing* (3.8.3). One of the only alternatives to extracting information from within Prefetch files is *Advanced Prefetch File Analyse* (AFPA). One of the issues within AFPA is its lack of ability to extract specific file entries from within Prefetch files. As discussed within *Current Solutions* (2.9) this is a problem as the Prefetch files may contain hundreds or thousands of entries causing a forensic examiner to search through them manually. An alternative use for WMP Extractor would be the ability for a forensic examiner to enter the file extension of which they wish to look for within the Prefetch files e.g. .doc, WMP Extractor would then return results relating to the specified file extension i.e. .doc.

WMP Extractor parses the WMP database and WMP Prefetch file on a hexadecimal level looking for specific patterns discovered within *Overview of Current Database\_372.wmdb* (section 2.4). Another use for WMP Extractor would be the use of it as a monitoring tool. Windows® Server keeps error logs for items such as logon attempts, both passed and failed logon attempts are recorded within the error logs. Error logs upon Windows® Server require the systems administrator to check them to see if there have been any failed logons which could signify someone trying to illegally enter the system by guessing the password. WMP Extractor could be designed to scan the error logs upon the server and look for failed login attempts, once a failed login attempt has been found a message would then be sent to the systems administrator alerting them. To enable WMP Extractor to perform as a monitoring tool the current WMP Extractor software would need to reside upon the server where it would periodically check error logs, this part of WMP Extractor would act as the sever. A client would need to be created that would run on the systems administrators computer, the client would receive messages from the server across the network informing the systems administrator of any predefined criteria such as failed login attempts.

WMP Extractor is able to analyse the WMP database and WMP Prefetch files and return the results within minutes. With ever increasing storage capability and decreasing price, computers have been shipping with increasing storage capacity. Computer forensic examiners may have a substantial case load at any one time (Philip, A. Cowsen, D. David, C 2009). With the ever increasing storage capabilities and the large caseloads, forensic examiners need to prioritise cases to ensure time,

money and resources are put to best use (Philip, A. Cowsen, D. David, C 2009). To enable a forensic examiner to better prioritise time, money and resources WMP Extractor could be used as a triage tool. Triage tools allow the software to be run upon a suspect's machine with little impact and allow the forensic examiner to access data almost immediately. With the ability to access data almost immediately the forensic examiner is able to make a choice to whether that machine requires further investigation or is of high priority if data returned is of interest (Christy, J, 2007). After WMP Extractor has finished the analysis of both the WMP database and WMP Prefetch files a report with any results found is populated. If the results produced by WMP Extractor contain information that is relevant to the case e.g. if the suspect has been watching illegal media then the computer can be seized. Another example would be that if there are several machines to be examined by the forensic examiner the results may help prioritize in which order machines are examined in, the results may show the machine is of higher priority if results from the WMP database or WMP Prefetch are found to contain illegal media or de-prioritize if no relevant results were found. The use of WMP Extractor as a triage tool will allow forensic examiner to save time, money and resources for the reasons mentioned above.

#### 4.4.2. Further Work into WMP Extractor

Below will discuss the further work that could be carried into the progression of WMP Extractor, it will discuss areas such as improvements as well as future features that would be beneficial.

People use words to navigate to websites such as *www.northumbria.ac.uk* rather than their IP address as names are easier to remember, it could be argued that people do the same by naming their files using a naming rather than number convention. The naming of files allows users to distinguish between two different documents without the need to open them. If people name their files to coincide with what they contain it would allow for a search function to be built into WMP Extractor. Two types of search mechanisms could be implemented into WMP Extractor. The first is once WMP Extractor has performed an analysis of either the WMP database or WMP Prefetch or both, the forensic examiner could search the report for the keywords and return any matches. The second search function would allow the forensic examiner to populate a list with search terms. During the analysis, file entries recovered from the WMP database or WMP Prefetch could be checked in real time, any positive results to be returned to the screen to provide feedback to the forensic examiner.

Arguably EnCase (Guidance, 2011) is one of the biggest pieces of forensic software used within computer forensic industry. One of the features of EnCase is its ability to bookmark items. The bookmarking facility allows a forensic examiner to create a folder structure and store bookmarks within that link to where the evidence is stored within the case file. The ability to bookmark items allows the forensic examiner to organise evidence, once evidence has been bookmarked it also allows the examiner to negate the need to search through the case file to find the evidence again. Bookmarks within EnCase work similarly to how bookmarks within web browsers work, they contain a link to the actual website its self. Introducing a bookmarking facility within

WMP Extractor would enable a forensic examiner to add entries extracted from the WMP database and WMP Prefetch to a central location, the forensic examiner would be able to return to the bookmarks at a later date and not have to search through the possible thousands of search results.

As discussed *Initial Research* (2.3) there are very few current solutions to the extraction of media viewed using WMP across the Windows® Operating System. While WMP Extractor was developed because there is currently no known software application available for the extraction of media viewed using WMP upon Windows® 7. While WMP Extractor is able to extract information about media viewed using WMP upon Windows® 7 it is unable to the same analysis on previous versions of WMP. Future research into previous versions of WMP databases would provide the data such as that detailed in *Overview of CurrentDatabase\_372.wmdb* (2.7) and *Overview of WMPAYER.EXE-xxxxxxx.pf* (2.8) to construct WMP Extractor to perform an analysis on multiple versions of Windows®. With the ability to perform analysis upon different versions of Windows® about media viewed using WMP, it would provide the forensic examiner with more comprehensive tool.

Windows® 7 (as well as other versions of Windows®) records information about external storage devices plugged in i.e.USB flash drives, this information is stored within the Windows Mounted Devices within the registry. The information contained within the Windows Mounted Devices includes the serial number for the external storage device as well as the drive letter that was assigned to it when it was mounted within Windows®.If a media file was viewed using WMP and reordered in with the WMP database or WMP Prefetch from an external storage device the presence of the external media would be recorded within the Windows Mounted Devices. The drive letter assigned to the external storage device would be reflected within entry of either the WMP database or WMP Prefetch files. WMP Extractor could be implemented to recover the external storage device(s) serial number and drive letter and match these with the same drive letters reordered within the WMP database and WMP Prefetch files. The combining of both sets of information from the Windows Mounted Devices as well as the WMP database and WMP Prefetch would allow the forensic examiner to have a better understanding of what the suspects have been doing.

When performing a live analysis upon a computer, a forensics investigator may want to make a copy of both the WMP database and the Windows™ Prefetch file. These may want to be kept for future analysis if future developments show hidden data stored within the database that is not currently known. To enable this feature when a forensic examiner runs the program using the default file locations it should ask them if they would like to copy the files to the file location of where the program is being run from. When copying the files over it would first need to create an MD5 hash value before the files are copied over and then after, it would then need to compare the hash values to ensure that they match for evidential integrity.

While the default file locations for both the WMP database and the Windows™ Prefetch are consistent throughout each version of Windows™ 7 it is possible for the user to change the file directory's or move the files. One way to get around this would



be when running the program in default file location mode and it discovering the files do not exist, would be for the forensic examiner to be given the option to search the entire drive. If a user has enough technical knowledge they may choose to rename the files and/or the file extensions. One way to overcome this issue would be similar to searching the entire hard drive but examining each file's signature, compare it to the known file signature both the WMP database and Windows® Prefetch.

As mentioned within *Background to Windows Media Player* (2.1), WMP version 12.0 is capable of viewing more different types of media than any other version of WMP. Not only is version 12.0 of WMP player able to view the most varied media types, it is possible to download a codec pack that enable it to view more (Download, 2011). While only the main media types have been included for extraction from the WMP database and Windows™ Prefetch files it would be beneficial if WMP Extractor was able to recognise as many different formats as possible.

#### 4.5. Conclusions

As with many other versions of Windows® including Vista, XP and 2000, WMP within Windows® 7 is the default media viewer. At the completion of this project Windows® 7 is still the current Operating System from Microsoft® with the next version not expected until 2012 (InfoWorld, 2010). Research conducted by the author after this project shows that there is still no known solution to the extraction of information about media viewed using WMP within Windows® 7, although companies such as Guidance Software may be conducting research, for commercial reasons they have not released information regarding it (Guidance Software, 2011). The project aimed at quickly identifying what, and if illegal media had been viewed in the hope of reducing the national backlog by prioritising the case load. While it may not be possible to prevent suspects from viewing illegal media, the use of WMP Extractor combined with its continued development will assist forensic examiners in the quick identification of those who have.

After initial investigation into both the WMP database and WMP Prefetch files, patterns were found to exist prior, throughout and after media entries that resided within both files. It was established that information stored within them included:

1. File name
2. Full file directory
3. Information from the “Details” section of the file properties
4. Size of file in bytes (not ‘Size of disk’)
5. Size of file in pixels
6. Date added to WMP
7. Date last viewed
8. Number of times viewed

While there are many other multimedia viewers the suspects can use to view media, it is safe to assume that WMP is the most widely installed media player due to its presence within most Windows® Operating Systems and the possibility of being the most used with being the default media player. A solution was proposed to counter the current lack of software that enables a forensic examiner to see what media a suspect has been viewing using WMP, the solution that was devised was WMP Extractor. The aims of WMP Extractor were to extract information from the WMP database and WMP Prefetch files, output that information to a report while maintaining evidential integrity and provide the computer forensic examiner with a GUI to conduct the analysis from.

To ensure that WMP Extractor met the aims of the project testing was carried out on both the fitness for purpose and build quality. The first aim of the project was to extract information from both the WMP database and WMP Prefetch files while maintaining evidential integrity. Test carried out on WMP Extractor used test data that was created specifically for this project. The testing of WMP Extractors ability to extract information from the WMP database and WMP Prefetch then produce a report concluded that, it is able to do so accurately while maintaining evidential integrity. While WMP Extractor is a proof of concept it was decided that the use of a GUI would require less training for the examiner to undertake prior to using WMP Extractor, the GUI would also enable more control over what the examiner can do during the analysis of the WMP database and WMP Prefetch files. To test the effectiveness of WMP Extractor five computer forensic students were asked to perform an analysis of the WMP database and WMP Prefetch files then asked for feedback, while some recommendations were made to the GUI overall it is effective. Overall all areas of testing of WMP Extractor proved successful, while there are areas that need improving upon such as user feedback during an analysis, tests showed that WMP Extractor has the potential to be a beneficial tool to the computer forensic community.

During the research into how WMP stores information about media viewed within *Overview of CurrentDatabase\_372.wmdb* (2.7), patterns were discovered prior to, during and after a media file entry within the WMP database. While the information stated above was found within the WMP database, it is suspected that more information may be available from within the database. Due to the time frame of this project, no known research into the area of how WMP stores information within the WMP database and no information released from Microsoft on the subject it is not known what other information may be present within the database. The tools and techniques identified within *Tools and Techniques* (2.12) were chosen as they were deemed to be the best way to design and implement WMP Extractor. The tools and techniques used within the construction of WMP Extractor were sufficient within the scope of this project, they provided the ability to design and construct WMP Extractor so that it met all requirements as discussed within *Evaluation of Fitness for Purpose* (4.1. While the tools and techniques used were sufficient within the scope of this project, should the project be expanded upon as suggested within *Further Work*(4.4) these would need to be re-evaluated.

Limitations within this project have a varying degree on the overall effectiveness of WMP Extractor. As discussed within *Software Requirements and Justification* (2.11) Java was chosen as programming language in which WMP Extractor was constructed in. As discussed within *Further Work* (4.4) the main disadvantage to using Java is that it requires the JVM to be installed, without the JVM WMP Extractor will not run. While as mentioned the forensic examiner would be able to perform a static analysis upon a machine that has the JVM installed, the inability to perform a live analysis reduces WMP Extractor usefulness. To overcome this limitation; the use of a different programming language that is not reliant on the JVM or other third party software was suggested and C# was deemed to meet this criteria (Microsoft Corporation, h 2010). Another limitation identified within WMP Extractor is the very narrow scope on which it is aimed; it is currently only able to extract information from the WMP database upon Windows® 7. While this was by design due to research suggesting that the extraction of information about media viewed; using WMP within Windows® 7 had not been covered by the computer forensic community, it does limit the application to only computers running Windows® 7. Suggested within *Further Work* (4.4) is that to overcome this limitation the continued development into research of other versions of the WMP database and, implementation of this research into WMP Extractor should be continued.

While there are limitations to WMP Extractor these are expected due to either the scope of the project or the tools used in its construction. Overall WMP Extractor has met all the requirements laid out within *Software Requirements and Justification* (2.11), this project has also been completed within the time frame set out prior to the project initiation in the *Terms of Reference* (5.1). With both of these requirements met it can be argued that the project including WMP Extractor can be considered a success. While WMP Extractor has been a success within the scope of this project it is by no means finished. It is hoped that now the computer forensic community has the ability to extract information about media viewed using WMP within Windows® 7 it will be built upon as discussed within *Further Work* (4.4). It is hoped that this project and its outcomes will go some way towards reducing the national backlog of cases that computer forensic examiners face.

It should be noted that while not within the scope of this project both Windows® XP and Windows® Vista record entries within the WMP Prefetch files the same as Windows® 7. WMP Extractor was tested with several Prefetch files obtained from both XP and Vista in which media entries were contained. WMP Extractor is capable of analysis both Windows® XP and Vista WMP Prefetch files. While more testing would be required to ensure evidential integrity is maintained and that results produced are accurate, preliminary results are encouraging.

#### 4.6.Recommendations

The purpose behind the idea and development of WMP Extractor has been to allow forensic examiners to extract information about media viewed using WMP within Windows® 7. WMP Extractor although a proof of concept has shown that it is possible to identify and extract information from Windows® 7. Principles applied

within this project can be applied to other areas that would assist in the gathering of information of media viewed of which have not been included within the scope of this project. Testing of `WMP_Extractor` has shown that although it conforms to the requirements set out in *Software Requirements* (2.11), there is still room for future improvement as show in *Further Work* (4.4).

Whilst many future improvements to `WMP_Extractor` have been set out within *Further Work* (4.4), the beneficial improvements to `WMP_Extractor` in the short term are listed below.

- The ability to read previous WMP database files
- The ability to search through entries returned
- The ability to retrieve information about external media
- Develop `WMP_Extractor` as an 'EnScript' for use within EnCase

As discussed to ensure the value of `WMP_Extractor` is maintained within a forensic role, its continued development is key. Microsoft® releases updates for Windows® (that are currently supported) every Tuesday; known as 'patch Tuesday'. If an update changes the pattern of the known file entry headers that are present prior to an entry within the WMP database or WMP Prefetch, it would render `WMP_Extractor` unusable. Continued testing development after any updates that may affect the known patterns within the WMP database or WMP Prefetch will mitigate the issue.

The use of `WMP_Extractor` extends beyond the scope of this project. Principles behind `WMP_Extractor` are able to be manipulated into serving not only areas within the computer forensic community, but outside the focus of forensic software. In the terms of applying the basic principles of `WMP_Extractor` outside the scope of forensics software, its use can be used within the auditing of files. The basic function of `WMP_Extractor` is to search through files using Hex looking for known patterns or clusters. `WMP_Extractor` could be changed to allow it to check not for known patterns but to check the state of specific areas within a file or database, by doing so it will allow `WMP_Extractor` to compare known values against current values that reside within a file. Any anomalies that are found can then be reported back to the user. The same principles that could be used in the auditing of files would also apply to checking the integrity of files. Once again `WMP_Extractor` could be used to ensure that no unauthorised changes have been made to a file or database. While it could be argued that this could be done using the MD5 hash algorithm to check for any changes, certain parts of the file or database may be allowed to be updated which would change the MD5 hash value.

`WMP_Extractor` has shown the potential for further development as well as use in other areas. Depending on `WMP_Extractor's` future will depend on the recommendations and thus the required work. In the short term it is recommended that

further tests be carried out within the submission to either the CFTT or BSI, as discussed within *Evaluation of Project Process* (4.3). Further testing will help ensure that the results produced are valid within court due to the reputable nature of the CFTT and BSI. Further short term recommendations include the analysis of other versions of WMP database. As discussed within *Background to Windows Media Player* (2.1) WMP comes with almost all versions of Windows®. While Windows® 7 is currently the latest version available computer forensic examiners will still need to analyse legacy system which may contain different versions of WMP. All of these recommendations including the development of WMP Extractor as a triage forensic tool will maximise the role in which it was designed for.

## **5. Appendix**

### 5.1. Terms of Reference

#### **Project Terms of Reference**

**Module:** Individual Project (CM0645)

**Name:** James Silman

**ID:** 08018306

**Course:** Computer Forensics

**Title:** The Identification of Media Viewed Using Windows Media Player within Windows 7 Through the use of Cluster Analysis  
**Supervisor:** Dr. Christopher Laing

**Second Marker:** Prof. Ahmed Bouridane

**Project Type:** General Computing

#### 5.1.1.1. Project title

The Identification of Media Viewed Using Windows Media Player within Windows 7 Through the use of Cluster Analysis

#### 5.1.1.2. Background to project

Windows Media Player (WMP) in Windows® 7 now comes with more codecs<sup>8</sup> than any other versions of Windows® previously released, this enables WMP to view a larger number of file types. This combined with WMP being the default program to play media files within Windows® 7 leads to a greater chance of it being used to view illegal media. The WMP database can contain file name, file properties, music, video and photo.

This project has been chosen for three reasons. Microsoft® Windows® holds a dominate share of the operating system market in both the home and corporate environment (6). All new and several past releases of Microsoft® Windows® Operating System come preinstalled with WMP. As mentioned this increases the chance that it will be used to view illegal media as users will not be required to install other media player to view media thus providing a rich source of information for a forensics examiner.

This project has been chosen because there is no current forensic tool that is able to extract information from the WMP database in Windows® 7 that stores metadata about the media viewed. There is a version that is able to extract data from the database created by version 9 of WMP but is not compatible with other versions.

While the registry records the last several played files in the registry it is possible to turn this feature off from within WMP, while it is not possible to stop WMP recording what has been played in the database (4). If the forensic examiner wants to extract the metadata from the database file it would require them to manually go through with a hex editor which could be very time consuming as there may be hundreds or possibly thousands of entries inside the database. The WMP database file is not dynamic so even if media is viewed and then removed the information will remain within the database indefinitely. This is extremely useful as if media is viewed using CD's or memory sticks there will be a log allowing for the possibly showing that the media was viewed on that machine if the illegal material is found separate to the computer.

The main objective of the project is to create a piece of software that will extract information from within the Windows Media Play (WMP) database file `CurrentDatabase_372.wmdb` from within "C:\Users\%USER%\AppData\Local\Microsoft\Media Player" on Windows® 7 (3).

From the initial research conducted it has shown that the `CurrentDatabase_372.wmdb` stores the complete file path of the media viewed using WMP i.e. *D:\illegal media\videos\illegal video.mpg*.

---

<sup>8</sup> Codecs are a plug-in for Windows Media Player that allow it to play different media types.



The software tool developed will be expected to read data from several locations upon the Windows 7 operating system to extract as much data as possible about the media viewed using WMP. The information that is collected by the software tool must then output it into an understandable format such as a report. The program should output the results to a Graphical User Interface (GUI) for ease of interpretation.

The software tool to be created will need to process possibly large amounts of data efficiently while using as least amount as possible of processing power and in a timely fashion. This will be achieved by using the built in Application Programming Interface (API) that will scan the files using a Hex scanner as it will more than likely be the most efficient algorithm.

To ensure that the software tool behaves as it should a large amount of tightly controlled test data will be loaded into a Windows 7 test environment. This will demonstrate the accuracy of the software tool when reporting back to the end user and will also highlight its strengths and weaknesses thereby allowing it to be improved. The main aim of the testing is to show that the data recovered by the software tool has not been altered in anyway thereby comply with the first ACPO guideline.

#### 5.1.1.3.Aims of the project

- The aims of this project are:
- To investigate what information is held on a Windows® 7 operating system with reference to any media that has been viewed using WMP.
- To create a software tool that will extract information about the files that have been viewed using WMP within Windows® 7 Operating System.

#### 5.1.1.4.Objectives

- To investigate what information is stored within the WMP database on Windows® 7.
- To research if other sources of information are linked to the WMP database that is stored within Windows® 7.
- To identify an appropriate programming language.
- To create a software tool to extract the information from within the files found to contain relevant data.
- To develop and appropriate testing strategy for the software tool to be created.
- To evaluate the project as a whole covering analysis through to the conclusion, the software tool and to produce a final report.
- Identify areas that require further research.

#### 5.1.1.5.Ethical issues

The software tool to be created will provide a text based report of files viewed using WMP in Windows®7, it has the possibility to be misused if it is used out of context or without the users permission.

Overall this project raises no ethical issues as all test data will be created, as well as no third parties being involved for there to be any data protection issues. It is possible for the software tool to be abused if it is used outside of an examination

#### 5.1.1.6.Relationship to course

##### Object Orientated Programming (OOP) CM0551

Programming will have the most effect as it will help contribute towards a large proportion of this project with creating the software tool. It has set up not only the ability to use Java within this project to create the software tool required it also has had the advantage learning the OOP concept for use with another language if necessary.

##### Computer Forensics CM00431, CM0664, CM0541

Computer forensics will help with the research into the files initially using software such as EnCase to discover what evidence is available to recover. It will also help with making sure the research project complies with the Association of Chief Police Officers (ACPO) to make sure that the results obtained by the software are admissible in court.

##### Professional Development & Project Management CM0558

Professional development and project management will help with the creation of the report for this project, it will enable the effective use of time to ensure all aspects of the project are completed on time, as well ensuring the correct format is used for the report and all references are correctly cited.

##### Professionalism & Ethical Practice CM0648

Professionalism and ethical practice will contribute to this report by ensuring that all aspects of the project remain ethical which will in turn allow this project to remain professional in its self.

##### Relational Databases CM0429

Relational databases are relevant as once the data has been extracted from Windows® 7 it will be stored inside a database that will allow for the easy viewing and searching of information.

#### 5.1.1.7.References/Bibliography

1. GEEP EDS LLC (2010) Darik's Boot And Nuke [online] Available at: <http://www.dban.org/>[Accessed 18 October 2010]
2. Microsoft Corporation (2010) Windows® 7 [online] Available at: [Http://www.microsoft.com/windows/windows-7/compare/starter.aspx](http://www.microsoft.com/windows/windows-7/compare/starter.aspx)[Accessed 18 October 2010]
3. Microsoft Corporation (2010) Windows® 7 [online] Available at: <http://support.microsoft.com/default.aspx?scid=kb;en-us;Q272116> [Accessed 18 October 2010]

4. Microsoft Corporation (2010) Windows® 7 [online] Available at: <http://support.microsoft.com/kb/243621>[Accessed 18 October 2010]

6. Net Market Share (2010) Operating System Market Share [online] Available at: <http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8>[Accessed 18 October 2010]

#### 5.1.1.8.Resources

- Independent Development Environment (IDE)

This will be used to create the software tool required by this project. There are several IDE's available in Pandon Basement for an appropriate language to be used.

- Microsoft® Windows® 7

Several operating systems will be used to check the effectiveness of the software tool created as each operating system may construct files differently. The creator of this project owns all licences for these operating systems.

- Computing, Engineering and Information Science (CEIS) Image

The normal CEIS lab image will provide all the necessary software applications.

- USB Memory Sticks

These will be used to view media off to create entries with several files that are not just local. This will allow the effective testing of the software tool to be created by this project. The creator of this project owns several memory sticks that will be securely erased before being used to prevent cross contamination.

#### 5.1.1.9.Structure/contents of project report

- List of contents
- Abstract
- Introduction
- Analysis
  - Investigation into `CurrentDatabase_XXX.wmdb`
  - Investigation into `WMPLAYER.EXE-xxxxxxxx.pf`
- Description of the problem
- Literature survey
- Identification/justification of detailed requirements

- Discussion of possible approaches/technologies
- Synthesis
  - Design
  - Implementation
  - Testing
- Evaluation/Conclusion
  - Product evaluation
  - Project evaluation
  - Conclusion/recommendations
- References
- Bibliography
- List of Appendices
  - Appendix 1: Terms of Reference
  - Appendix 2: Ethics form
  - Appendix 3: As required

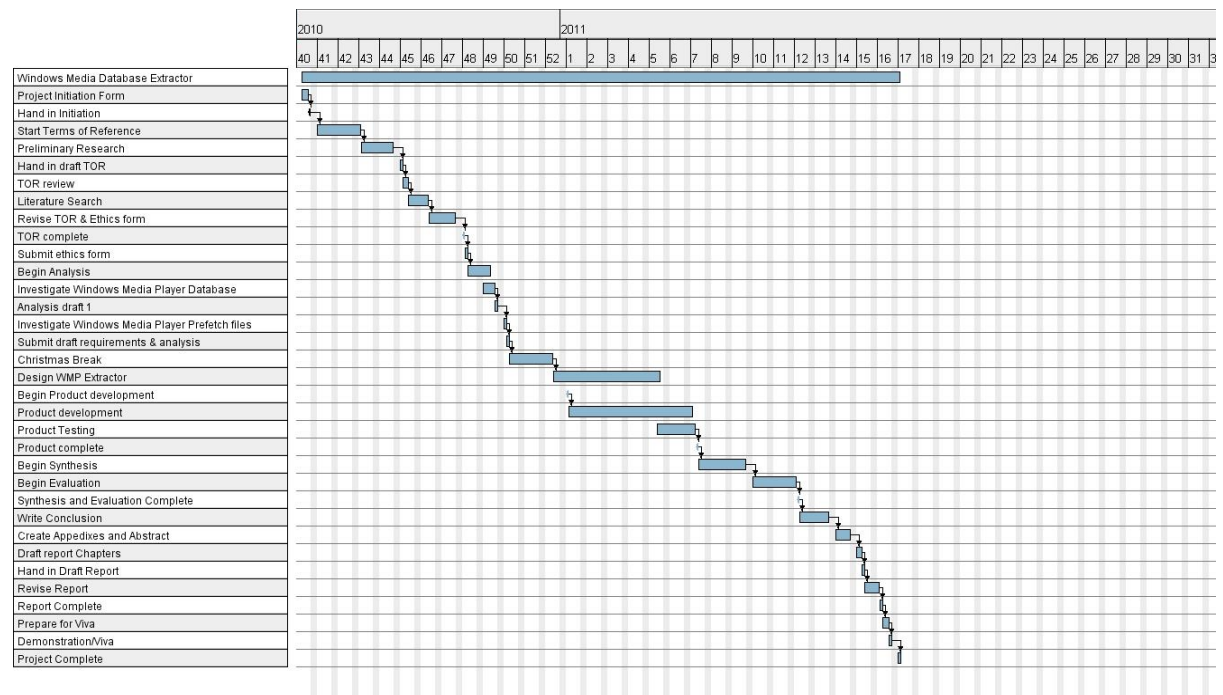
#### 5.1.1.10. [Marking scheme](#)

This project will fall under the general computing project marking scheme.

#### 5.1.1.11. [Appendices](#)



## 5.1.1.11.1. Appendix 1: Project plan



## 5.2. Ethics Form

**School of Computing, Engineering and Information Sciences**

StudentProject/Dissertation

EthicsRegistrationForm – B

For Undergraduate and Postgraduate Programmes

<b>Student's name:</b>	James Silman		
<b>Supervisor's name:</b>	Dr. Christopher Laing		
<b>Project Title:</b>	The Identification of Media Viewed Using Windows Media Player within Windows 7 Through the use of Cluster Analysis		
<b>Programme:</b>	Computer Forensics		
<b>Project Module Code:</b>	CM0645	<b>Academic Year:</b>	2011
<b>Date of commencement:</b>	October 2010		

<b>Ethical considerations in the research project</b>	
1. Does your project require external ethical approval or police Criminal Records Bureau (CRB) clearance (e.g. work with NHS personnel or on NHS premises; or work in Schools or Institutions with vulnerable people)?	No
2. Do any participants constitute a 'vulnerable group' (e.g. elderly people or under 18 years of age)?	No
3. Will the research involve sensitive information of (a) a commercial nature? (b) a personal nature? (c) a political nature?	No No No
4. If yes to any of 1 to 3, above, what steps have you taken to deal with the issue?	
5. Are there likely to be any risks/harm for you, the university or for the participants in your project? If so, what are these risks?	No
6. Is there any potential for misuse, or unintended use, of a developed artefact, or potential to break the law? If so, what is the potential and the likely degree of risk?	No
7. If yes to any of questions 1- 6, above, have appropriate measures been detailed in your Proposal/TOR?	No

**Statement by Student**

I have read the Northumbria University Research Ethics and Governance Handbook 2007-2008: (<http://northumbria.ac.uk/researchandconsultancy/sa/ethgov/?view=Standard>)



(located in the E-learning Portal – CEIS School Office Customer Support) and confirm that the answers I have given above are correct. Where further issues arise \* under items 1 - 6 [above] I will discuss these and obtain agreement from my supervisor to proceed. I understand that if I am interviewing people I must provide a Project information sheet (CEIS Form F), inform the participants about my research and obtain their consent using the standard consent form (CEIS Ethics Form C). I have also described in writing (to be submitted to my supervisor as an appendix to this form) how I intend to approach these issues in the research. **This form is auditable and must be filled in by all students and supplied together with consent forms and any appendices to your supervisor. Students must also supply a copy of the forms (B and C if applicable) to the CEIS Research and Projects (Ethics) Office (Ellison Building, D001).**

Students Signature: ..... Date: .....

Supervisor's signature: ..... Date:.....

## 6. References

### A

Abras, C., Maloney-Krichmar, D. And Preece, J.(2004) ‘User-Centered Design’, *Encyclopedia of Human-Computer Interaction*, pp.1-14.

ACPO (2010) *Good Practice Guide for Computer-Based Electronic Evidence (v. 4.0.)* London, UK: 7Safe.

### B

Bruegge, B. Dutoit, A.H. (2009) ‘Use Case Diagrams’, *Object-Oriented Software Engineering Using UML, Patterns, and Java*, 3<sup>rd</sup> edition, pp.31

British Standards Institute, a (2011) BSI [online] Available at: <http://www.bsigroup.com/en/> (Accessed 21 March 2010)

British Standards Institute, b (2011) BSI Healthcare and Testing Services [online] Available at: <http://www.bsigroup.com/en/ProductServices/> (Accessed 21 March 2010)

British Standards Institute, c (2011) More About BSI Group [online] Available at: <http://www.bsigroup.com/en/About-BSI/About-BSI-Group/> (Accessed 21 March 2010)

British Standards Institute, d (2011) Direct Product Testing Services with BSI [online] Available at: <http://www.bsigroup.com/en/ProductServices/About-Testing/>

### C

Computer Forensic Tool Testing, a (2003) CFTT [online] Available at: <http://www.cfft.nist.gov/> (Accessed 21 March 2010)

Computer Forensic Tool Testing, b (2003) CFTT Methodology Overview [online] Available at: [http://www.cfft.nist.gov/Methodology\\_Overview.htm](http://www.cfft.nist.gov/Methodology_Overview.htm) (Accessed 21 March 2010)

Christy, J (2007) “Solutions for the Filed”, *Techno Security's™ Guide to E-Discovery and Digital Forensics*, Syngress, 1<sup>st</sup> Edition, pp.137

Cool Buster (2010) *How to delete Windows 7 Prefetch*[online] Available at: <http://www.coolbuster.net/2009/08/empty-delete-prefetch-windows-7.html> [Accessed 29 January 2011]

CNN (2004) *Microsoft Hit by record EU fine* [online] Available at: <http://web.archive.org/web/20060413082435/http://www.cnn.com/2004/BUSINESS/03/24/microsoft.eu/> (Accessed 26 November 2010)

CNET (2005) *Still 'no demand' for media-player-free Window*[online] Available at:  
[http://news.cnet.com/2100-1016\\_3-5960750.html](http://news.cnet.com/2100-1016_3-5960750.html) (Accessed 26 November 2010)

## D

Dariks Boot and Nuke (2010) *DBAN* [online] Available at:  
<http://www.dban.org/about> (Accessed 12 November 2010)

Deitel, P. Deitel, H, (2009) 'Pseudo code', *How to Program*, Prentice Hall, 7<sup>th</sup> edition, pp.111

Download (2011) *Media Player Codec Pack* [online] Available at:  
[http://download.cnet.com/Media-Player-Codec-Pack/3000-13632\\_4-10749065.html](http://download.cnet.com/Media-Player-Codec-Pack/3000-13632_4-10749065.html)  
(Last accessed March 4 2011)

## E

Eclipse (2011) *Eclipse* [online] Available at: <http://www.eclipse.org/> (Last accessed March 5 2011)

## F

Filesig Software Solutions (2010) *WMDB Extractor* [online] Available at:  
<http://www.simplecarver.com/tool.php?toolname=WMDB%20Extractor> (Accessed 29 November 2010)

Forensic Focus, a (2006) *Windows Media Player*[online] Available at:  
<http://www.forensicfocus.com/index.php?name=Forums&file=viewtopic&t=402&postdays=0&postorder=asc&start=0>  
(Accessed 29 November 2010)

Forensic Focus, b (2010) *Forensic Focus*[online] Available at:  
<http://www.forensicfocus.com/> (Accessed 29 November 2010)

## G

Grubb, P. Takang, A.A (2003) 'Fitness for Purpose', *Software Maintenance: Concepts and Practice*, World Scientific Publishing Co Pte Ltd, 2<sup>nd</sup> edition, pp.273

Guidance Software (2011) *EnCase* [online] Available at:  
<http://www.guidancesoftware.com/> (Last accessed 25 February 2011)

## H

Hay, S.A (2010) *Advanced Prefetch File Analyser*[online] Available at:  
<http://www.mitec.cz/>(Accessed 29 November 2010)

## I

Info World (2010) *What to expect from Windows 8*[online] Available at:  
<http://www.infoworld.com/d/windows/what-expect-windows-8-795?page=0,1>

J

JCreator (2010) *JCreator* [online] Available at:  
<http://www.jcreator.com/> (Accessed 8 March 2011)

## K

Karp, D.A. (2004) 'Keeping an eye on Prefetch' *Windows XP Annoyances for Geeks*, O'Reilly Media; 2<sup>nd</sup> edition, pp210

Kennedy, D. (2009) *Computer Crime Investigation*. [Lecturer at Northumbria University]

## L

Liang, Y.D. a (2010) 'GUI Basics', *Introduction to Java Programming: Comprehensive Version*, Prentice Hall, 8th edition, pp.6

Liang, Y.D. b (2008) 'GUI Basics', *IntroductionTo JAVA Programming: Comprehensive Version*, Pearson Education, 7<sup>th</sup> edition, pp.448.

## M

Market Share (2010) *Operating System Market Share* [online] Available at: <http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8> (Accessed 12 November 2010)

Microsoft® Corporation, a (2010) *Prefetch Folder* [online] Available at: <http://windows.microsoft.com/en-US/windows-vista/What-is-the-prefetch-folder> (Accessed 12 November 2010)

Microsoft® Corporation, b (2005) *Windows Version History* [online] Available at: <http://support.microsoft.com/kb/32905> (Accessed 12 November 2010)

Microsoft® Corporation, c (2009) *Windows 7 Release* [online] Available at: <http://www.microsoft.com/windows/windows-7/features/windows-media-player-12.aspx> (Accessed 12 November 2010)

Microsoft® Corporation, d (2009) *Windows 7 N editions offer choice* [online] Available at: <http://windows.microsoft.com/en-GB/windows7/products/What-is-Windows-7-N-edition> (Accessed 26 November 2010)

Microsoft® Corporation, e (2009) *Get Windows Media Player* [online] Available at: <http://windows.microsoft.com/en-us/windows/downloads/windows-media-player> (Accessed 26 November 2010)

Microsoft® Corporation, f (2010) *Windows Media Player 7* [online] Available at: <http://www.microsoft.com/presspass/press/2000/Jul00/WMP7PR.mspx> (Accessed 29 November 2010)

Microsoft® Corporation, g (2009) *Windows Media Player 11* [online] Available at: <http://social.technet.microsoft.com/Forums/en/w7itpromedia/thread/02cee7dc-1d67-4ad4-b807-70724c0558b0> (Accessed 29 November 2010)

Microsoft® Corporation, h (2011) *Visual C# Developer Centre*, [online] Available at: <http://msdn.microsoft.com/en-us/vcsharp/default> (Last accessed 1st March 2011)

Microsoft® Corporation, i (2011) *Dream Spark* [online] Available at: <https://www.dreamspark.com/default.aspx> (Last accessed 1st March 2011)

Microsoft® Corporation, j (2011) *Microsoft Windows SDK for Windows 7 and .Net Framework 3.5 SPI* [online] Available at: <http://www.microsoft.com/downloads/en/details.aspx?FamilyID=c17ba869-9671-4330-a63e-1fd44e0e2505&displaylang=en> (Last accessed 1st February 2011)

Microsoft® Corporation, k (2011) *Visual Studio 2010* [online] Available at: <http://msdn.microsoft.com/en-us/vstudio/aa718325> (Last accessed 1st March 2011)

Minasi, M, Mueller, J.P (2007) 'Windows Media Player' *Mastering Windows Vista Business: Ultimate, Business, and Enterprise*, John Wiley & Sons; Illustrated Edition, pp310

## N

National Institute of Standards and Technology (2011) About NIST [online] Available at: [http://www.nist.gov/public\\_affairs/nandyou.cfm](http://www.nist.gov/public_affairs/nandyou.cfm) (Accessed 21 March 2010)

NetBeans (2011) NetBeans IDE – The Smart Way to Code [online] Available at: <http://netbeans.org/features/index.html> (Last accessed 1st March 2011)

## O

Object Management Group (2011) *Getting Started with UML* [online] Available at: <http://www.uml.org/> (Accessed 1st March 2011)

Oracle, (2010) *Class MessageDigest* [online] Available at: <http://download.oracle.com/javase/1.4.2/docs/api/java/security/MessageDigest.html> (Last accessed 1st February 2010)

## P

Prefetch Information (2009) *Update Prefetch Parser* [online] Available at: <http://cfed-ttf.blogspot.com/2009/02/updated-prefetch-parser.html> (Last accessed 1st December 2010)

Philip, A. Cowsen, D. David, C (2009) "What is a Computer Forensic Laboratory", *Hacking Computer Forensic Exposed*, McGraw-Hill Osborne, 2nd edition, pp.42

## S

Sandler, C (2007) 'Hard Drives: bigger...' ,*Fix Your Own PC*, John Wiley & Sons; 8th Edition, pp.223

Stack Overflow (2011), Convert a String of Hex into ASCII in Java [online] Available at:

<http://stackoverflow.com/questions/4785654/covert-a-string-of-hex-into-ascii-in-java> (Accessed 12th January 2011)

Sobh, T (2010) 'Visualization of Large Software Projects by using Advance Techniques', *Innovations and Advances in Computer Sciences and Engineering*, Springer, 1<sup>st</sup> edition, pp.352

## T

The Telegraph (2009) *Microsoft Windows 7 smashes sales records*[online] Available at:<http://www.telegraph.co.uk/technology/microsoft/6513866/Microsoft-Windows-7-smashes-sales-records.html> (Accessed 29 November 2010)

## V

Video Lan (2010) *VLC Media Player* [online] Available at: <http://www.videolan.org/vlc/> (Accessed 12 November 2010)

## 7. Bibliography

Blogspot (2008) *Prefetch Information*[online] Available at:

<http://cfed-ttf.blogspot.com/2008/02/prefetch-information.html> (Accessed 29 November 2010)

Java, (2010) *The Java Language Specification*[online] Available at:

<http://java.sun.com/docs/books/jls/> (Last accessed 1st March 2011)

Java, (2011) *What is an Object?* [online] Available at:

<http://download.oracle.com/javase/tutorial/java/concepts/object.html> (Accessed 8 March 2011)

Regan, G.O. (2002) ‘A Practical Approach to Software Quality’, *Introduction to Software Quality*, Springer, pp.11

Ritchie, D.M (2006) *History*, [online] Available at:

<http://cm.bell-labs.com/who/dmr/> (Last accessed 1st March 2011)



## 7.1.Images

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF	
0x0F4690	2500	0000	08B9	00CA	E604	0800	0000	0865	%....'.Eæ.....e	1
0x0F46A0	5303	A39E	0000	75E8	0C10	0000	0000	0003	S.ëž..uè.....	2
0x0F46B0	0000	0004	0000	0000	0000	0000	0000	0000	.....	
0x0F46C0	0000	4AE7	B26F	CBB8	BB40	93F3	FAC5	F00F	..Jç'oE,»@"óúÅð.	
0x0F46D0	A203	0000	0000	0000	0000	0000	0000	0000	*.....	5
0x0F46E0	0000	BB12	0253	A627	FE46	ABBD	B953	9311	..»...S 'bF«'S".	
0x0F46F0	E97C	2F86	0003	639A	0000	D807	0000	0300	é /t...cš..0.....	3
0x0F4700	0000	0E00	0000	3073	CA83	DCDC	CB01	3073	.....0sEfÜÜE.0s	4
0x0F4710	CA83	DCDC	CB01	3200	0000	0000	0000	0000	ÊfÜÜE.2.....	6
0x0F4720	0000	0080	0700	0780	2A80	3100	3180	3580	...E...E*E1.1E5E	
0x0F4730	3900	3980	5900	5900	5900	5900	4465	7365	9.9EY.Y.Y.Y.Dese	7
0x0F4740	7274	0043	3A5C	5573	6572	735C	4A61	6D65	rt.C:\Users\Jame	8
0x0F4750	735C	5069	6374	7572	6573	5C44	6573	6572	s\Pictures\Deser	
0x0F4760	742E	6A70	6700	436F	7262	6973	006A	7067	t.jpg.Corbis.jpg	
0x0F4770	006A	7067	00A9	2043	6F72	6269	732E	2020	.jpg. Corbis.	9
0x0F4780	416C	6C20	5269	6768	7473	2052	6573	6572	All Rights Reser	
0x0F4790	7665	642E	0000	0000	0000	0000	0000	0000	ved.....	
0x0F47A0	0000	0000	0000	0000	0000	0000	0000	0000	.....	

(Figure 7 – shows the layout of information within the WMP database for an image entry) Please note that green was used to show the overlap of two segments of Hex.

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF	
0x0F5FE0	C001	0402	8001	0402	4001	0402	0001	0402	À...ë...@.....	
0x0F5FF0	C000	0402	8000	0402	4000	0402	0000	0402	À...ë...@.....	
0x0F6000	1889	7F83	1F05	5000	6000	0000	0000	0000	..f..P..`.....	
0x0F6010	0114	0200	0000	0000	C000	0000	0000	0046	.....À.....F	1
0x0F6020	4C00	0000	0114	0200	0000	0000	C000	0000	L.....À...	
0x0F6030	0000	0046	8301	2000	2000	0000	497E	0B69	...Ff. ....I~.i	2
0x0F6040	DCDC	CB01	497E	0B69	DCDC	CB01	574D	E87B	ÜÜE.I~.iÜÜE.WMè{	3
0x0F6050	4404	CA01	75E8	0C00	0000	0000	0100	0000	D.Ë.uè.....	4
0x0F6060	0000	0000	0000	0000	0000	0000	E200	1400	.....â...	
0x0F6070	1F44	471A	0359	723F	A744	89C5	5595	FE6B	.DG..Yr?\$D%ÅU·pk	
0x0F6080	30EE	2000	0000	1A00	EEBB	FE23	0000	1000	0í .....i»p#....	
0x0F6090	3081	E233	1E4E	7646	835A	9839	5C3B	C3BB	0 â3.NvFfZ~9\;Å»	
0x0F60A0	0000	AC00	3200	75E8	0C00	EE3A	102C	2000	...2.uè...i:.,.	
0x0F60B0	4465	7365	7274	2E6A	7067	0000	9200	0800	Desert.jpg...'	5
0x0F60C0	0400	EFBE	673E	A77B	673E	A77B	2A00	0000	..i%g>\$[g>\${*...	
0x0F60D0	2A07	0100	0000	2800	0000	0000	0000	0000	*.....{.....	6
0x0F60E0	4000	0000	0000	4400	6500	7300	6500	7200	@.....D.e.s.e.r.	7
0x0F60F0	7400	2E00	6A00	7000	6700	0000	4000	4300	t...j.p.g...@.C.	
0x0F6100	3A00	5C00	5700	6900	6E00	6400	6F00	7700	:\.W.i.n.d.o.w.	
0x0F6110	7300	5C00	7300	7900	7300	7400	6500	6D00	s.\.s.y.s.t.e.m.	
0x0F6120	3300	3200	5C00	5300	6100	6D00	7000	6C00	3.2.\.S.a.m.p.l.	
0x0F6130	6500	5200	6500	7300	2E00	6400	6C00	6C00	e.R.e.s...d.l.l.	
0x0F6140	2C00	2D00	3100	3000	3200	0000	1A00	0000	,.-.1.0.2.....	
0x0F6150	5100	0000	1C00	0000	0100	0000	1C00	0000	Q.....	
0x0F6160	2D00	0000	0000	0000	5000	0000	1100	0000	-.....P.....	
0x0F6170	0300	0000	2E6B	4A80	1000	0000	0043	3A5C	.....kJE.....C:\	8
0x0F6180	5573	6572	735C	4A61	6D65	735C	5069	6374	Users\James\Pict	
0x0F6190	7572	6573	5C44	6573	6572	742E	6A70	6700	ures\Desert.jpg.	
0x0F61A0	0028	0000	0009	0000	A01C	0000	0031	5350	.(.....1SP	
0x0F61B0	53E2	8A58	46BC	4C38	43BB	FC13	9326	986D	SâŠXF%L8C»ü."e~m	
0x0F61C0	CE00	0000	0000	0000	0060	0000	0003	0000	î.....	
0x0F61D0	A058	0000	0000	0000	006E	697A	6D6F	6E00	X.....nizmon.	9
0x0F61E0	0000	0000	0000	0000	00DE	B48E	1BAE	F4CA	.....P'Ž.øðÊ	

143

(Figure 8 - shows the layout of an image entry available later on the WMP database)

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF	
0x043580	3A00	0000	10F8	1308	4E03	B0FC	1004	0000	.....ø..N.°ü....	1
0x043590	0030	35C9	03A2	9E00	0001	0000	0001	0000	.05É.°ž.....	
0x0435A0	0050	C201	0085	6315	004D	5034	56F2	C053	.PÂ....c..MP4VòÀS	
0x0435B0	0555	0000	0000	0010	0080	0000	AA00	389B	.U.....€...².8>	
0x0435C0	714D	5034	5600	0010	0080	0000	AA00	389B	qMP4V....€...².8>	
0x0435D0	7100	0000	0000	0000	0000	0000	0000	0000	q.....	2
0x0435E0	0098	891D	0000	18E1	1500	0000	00D5	2517	.~%...á....Ö§.	
0x0435F0	00BD	3098	DBB3	3AAB	4F8A	371A	995F	7FF7	.¼0~Ô':«ÖŠ7.™_÷	
0x043600	4B00	0000	0000	0000	0000	0000	0000	0000	K.....	3
0x043610	00E2	4E31	3585	1600	4F82	F7AF	D645	FF04	.ân15....O÷~ÖEÿ.	4
0x043620	CE30	1FE2	5325	DCCB	01E0	1FE2	5325	DCCB	Ěà.àsšÜĚ.à.àsšÜĚ	
0x043630	0100	0000	0000	0000	0000	0000	0000	0000	.....	5
0x043640	0000	0000	0032	0000	0000	0000	0003	0000	.....2.....	
0x043650	0000	0000	0000	0000	0000	0000	0000	0000	.....	
0x043660	0000	0000	0000	0000	0030	35C9	03A2	9E00	.....05É.°ž.	
0x043670	0000	0000	0000	0000	0000	0000	0000	0000	.....	
0x043680	800C	000C	8026	0026	802A	802E	002E	002E	€...€&.€€*€.....	
0x043690	002E	002E	004E	6F72	7468	756D	6272	6961	....Northumbria	6
0x0436A0	0044	3A5C	5669	6465	6F73	5C4E	6F72	7468	.D:\Videos\North	7
0x0436B0	756D	6272	6961	2E61	7669	0061	7669	0061	umbria.avi.avi.a	8
0x0436C0	7669	0000	0000	0000	0000	0000	0000	0000	vi.....	

(Figure 9 - shows the layout of information within the WMP database for an video entry)

	0001	0203	0405	0607	0809	0A0B	0C0D	0E0F	0123456789ABCDEF	
0x102020	4C00	0000	0114	0200	0000	0000	C000	0000	L.....Ä...	1
0x102030	0000	0046	8301	2000	2000	0000	FE8A	1F49	...Ff. ....pŠ.I	2
0x102040	23DC	CB01	FE8A	1F49	23DC	CB01	346C	5D5B	#ÜĚ. pŠ.I#ÜĚ.4l] [	3
0x102050	1904	C901	0018	E115	0000	0000	0100	0000	..Ě...á.....	4
0x102060	0000	0000	0000	0000	0000	0000	E700	1400	.....ç.....	
0x102070	1F50	E04F	D020	EA3A	6910	A2D8	0800	2B30	.PàOB è:i.°ø...+0	
0x102080	309D	1900	2F44	3A5C	0000	0000	0000	0000	0 .. /D:\.....	
0x102090	0000	0000	0000	0000	0000	0050	0031	0000	.....P.1..	
0x1020A0	0000	0066	3E02	8B11	0056	6964	656F	7300	..f>.<..Videos.	5
0x1020B0	003A	0008	0004	00EF	BE54	3EBC	0A66	3E02	.....i%T>%f>.	
0x1020C0	8B2A	0000	0027	0000	0000	0001	0000	0000	<*...'. .....	
0x1020D0	0000	0000	0000	0000	0000	0056	0069	0064	.....V.i.d	
0x1020E0	0065	006F	0073	0000	0016	0068	0032	0000	.e.o.s.....h.2..	
0x1020F0	18E1	1516	39DA	2C20	004E	4F52	5448	557E	.á..9Ú, .NORTHU~	6
0x102100	312E	4156	4900	004C	0008	0004	00EF	BE66	1.AVI..L.....i%f	
0x102110	3E01	8B66	3E01	8B2A	0000	00C3	8E00	0000	>.<f>.<*...Äž...	
0x102120	0003	0000	0000	0000	0000	0000	0000	0000	.....	
0x102130	004E	006F	0072	0074	0068	0075	006D	0062	.N.o.r.t.h.u.m.b	7
0x102140	0072	0069	0061	002E	0061	0076	0069	0000	.r.i.a...a.v.i..	
0x102150	001C	0000	004D	0000	001C	0000	0001	0000	....M.....	
0x102160	001C	0000	0032	0000	0000	0000	004C	0000	.....2.....L..	
0x102170	0016	0000	0003	0000	00FC	B67F	8C10	0000	.....üq0Ě...	
0x102180	0035	3030	4742	0044	3A5C	5669	6465	6F73	.....D:\Videos	8
0x102190	5C4E	6F72	7468	756D	6272	6961	2E61	7669	\Northumbria.avi	
0x1021A0	0000	2800	0000	0900	00A0	1C00	0000	3153	..(.....1S	
0x1021B0	5053	E28A	5846	BC4C	3843	BBFC	1393	2698	PSâŠXF*L8C»ü."s~	
0x1021C0	6DCE	0000	0000	0000	0000	6000	0000	0300	mî.....`.....	
0x1021D0	00A0	5800	0000	0000	0000	6E69	7A6D	6F6E	. X.....nizmon	9
0x1021E0	0000	0000	0000	0000	0000	067D	63C3	C7BF	.....}cÄÇz	
0x1021F0	EC4B	8F12	FDAC	F21D	EA80	79A7	0047	6C46	iK .ý-ò.ê€yS.GlF	

(Figure 10- shows the layout of a video entry available later on the WMP database)



A	B	C	D	E	F	G	H	I	J	K	L	M
1	Examiner James	Case Number		1 Report Created	2011-03-09 19:10:59							
2		File Size (bytes)	Times Added	Times Last Viewed	Times Viewed	File Name	Directory Letter	Full Directory Path		Extra Details		
3												
4		MP3										
5												
6												
7		8414449 Sat Jan 29 18:04:39 GMT 2011	Sat Jan 29 18:04:55 GMT 2011		0 Kalimba	C:\		Users\Public\Music\Sample Music\Kalimba.mp3		mp3mp3A. Carthy and A. KingslowElectronicMr.		
8		4113874 Sat Jan 29 18:04:53 GMT 2011	Sat Jan 29 18:04:55 GMT 2011		0 Maid with the Flaxen Hair	C:\		Users\Public\Music\Sample Music\Maid with the Flaxen Hair.mp3		mp3mp3Claude DebussyClassicalRichard Stoltz		
9		4042585 Sat Jan 29 18:04:24 GMT 2011	Sat Jan 29 18:04:24 GMT 2011		0 Sleep Away	C:\		Users\Public\Music\Sample Music\Sleep Away.mp3		mp3mp3Robert R. AcrtiazzBob AcrtiBob		
10												
11												

(Figure 19 – shows audio entries extracted by WMP Extractor)

B	C	D	E	F	G	H	I	J
1								
2	File Size (bytes)	Times Added	Times Last Viewed	Times Viewed	File Name	Directory Letter	Full Directory Path	Extra Details
3								
4	JPG							
5								
6	879394 Sat Jan 29 18:04:26 GMT 2011	N/A	N/A	Chrysanthemum	C:\	Users\Public\Pictures\Sample Pictures\Chrysanthemum.jpg	Corbisjpgpg© Corbis. All Rights Reserved.	
7	845941 Sat Jan 29 18:04:26 GMT 2011	N/A	N/A	Desert	C:\	Users\Public\Pictures\Sample Pictures\Desert.jpg	Corbisjpgpg© Corbis. All Rights Reserved.	
8	595284 Sat Jan 29 18:04:26 GMT 2011	N/A	N/A	Hydrangeas	C:\	Users\Public\Pictures\Sample Pictures\Hydrangeas.jpg	Amish PateljpgpgMicrosoft Corporation	
9	775702 Sat Jan 29 18:04:26 GMT 2011	N/A	N/A	Jellyfish	C:\	Users\Public\Pictures\Sample Pictures\Jellyfish.jpg	Hang QuanjpgpgMicrosoft Corporation	
10	780831 Sat Jan 29 18:04:26 GMT 2011	N/A	N/A	Koala	C:\	Users\Public\Pictures\Sample Pictures\Koala.jpg	Corbisjpgpg© Corbis. All Rights Reserved.	
11	561276 Sat Jan 29 18:04:26 GMT 2011	N/A	N/A	Lighthouse	C:\	Users\Public\Pictures\Sample Pictures\Lighthouse.jpg	Tom AlphinjpgpgMicrosoft Corporation	
12								
13								

(Figure 20 - shows image entries extracted by WMP Extractor)

A	B	C	D	E	F	G	H	I
1	Examiner James	Case Number		1 Report Created	2011-03-30 13:04:30			
2		File Size (bytes)	Times Added	Times Last Viewed	Times Viewed	File Name	Directory Letter	Full Directory Path
3								Extra Details
4		AVI						
5								
6								
7		366983168 Wed Apr 06 17:00:42 BST 2011	N/A	N/A	House S01E01 Pilot	D:\	Videos\House S01E01 Pilot.avi	aviavi
8		366985216 Wed Apr 06 17:00:42 BST 2011	N/A	N/A	House S01E02 Paternity	D:\	Videos\House S01E02 Paternity.avi	aviavi
9		366979072 Wed Apr 06 17:00:42 BST 2011	N/A	N/A	House S01E03 Occam's Razor	D:\	Videos\House S01E03 Occam's Razor.avi	aviavi
10		367073280 Wed Apr 06 17:00:42 BST 2011	N/A	N/A	Northumbria	D:\	Videos\Northumbria.avi	aviavi
11								

(Figure 21 - shows video entries extracted by WMP Extractor)

## 7.2. Use Case Descriptions

<b>Use Case</b>	Extract Media Entries From WMP Prefetch Files
<b>Summary</b>	WMP Extractor scans the WMP Prefetch files for media entries and extracts them out and stores them in the report.
<b>Actor</b>	User
<b>Trigger</b>	WMP Extractor finds the WMP Prefetch files via the default file path or a manually selected file path
<b>Primary Scenario</b>	<ol style="list-style-type: none"> <li>1. PrefetchSearch find the WMP database via the given path</li> <li>2. PrefetchSearch receives file information from PrefetchSignatures</li> <li>3. PrefetchSearch searches for known file signature</li> </ol>
<b>Alternative Scenario</b>	1. PrefetchSearch is unable to find any audio entries
<b>Exceptional Scenario</b>	1. PrefetchSearch is unable to find any WMP Prefetch files
<b>Pre-Conditions</b>	WMP Extractor is running
<b>Post-Conditions</b>	Media entries copied out of database and stored within report
<b>Assumptions</b>	Valid name, case number/name and file name entered

(UC Description: Extract Media Entries from WMP Prefetch Files)

<b>Use Case</b>	Check MD5 hash value of WMP database and WMP Prefetch files
<b>Summary</b>	HashValue receives file location and file name from WMDBExtractor and PrefetchSearch
<b>Actor</b>	User
<b>Trigger</b>	File path and file name received from WMDBExtractor and PrefetchSearch
<b>Primary Scenario</b>	<ol style="list-style-type: none"> <li>1. File path and file name used to find file</li> <li>2. MD5 hash value calculated</li> </ol>
<b>Alternative Scenario</b>	PrefetchSearch is unable to find any audio entries
<b>Exceptional Scenario</b>	PrefetchSearch is unable to find any WMP Prefetch files
<b>Pre-Conditions</b>	WMP Extractor is running
<b>Post-Conditions</b>	MD5 hash value stored within report
<b>Assumptions</b>	WMDBExtractor and PrefetchSearch send file name and file path to HashValue

(UC Description: Check MD5 hash value of WMP database and WMP Prefetch files)



<b>Use Case</b>	Create GUI and Close WMP Extractor
<b>Summary</b>	GUI presented to examiner to allow for file location to be selected
<b>Actor</b>	User
<b>Trigger</b>	Examiner selects file location using the GUI
<b>Primary Scenario</b>	<ol style="list-style-type: none"> <li>1. Examiner selects file location using GUI</li> <li>2. File location passed to WMDbExtractor and PrefetchSearch</li> <li>3. WMP Extractor Extracts media entries from the WMP database and WMP Prefetch files</li> <li>4. Report is created with media entry information</li> </ol>
<b>Alternative Scenario</b>	<ol style="list-style-type: none"> <li>1. WMP Extractor finds WMP database and WMP Prefetch in given location</li> <li>2. No media entries are found to be located within the WMP database or WMP Prefetch files</li> </ol>
<b>Exceptional Scenario</b>	<ol style="list-style-type: none"> <li>1. File location selected does not contain the WMP database or WMP Prefetch files</li> </ol>
<b>Pre-Conditions</b>	WMP Extractor is running
<b>Post-Conditions</b>	WMP database and WMP Prefetch files analysed
<b>Assumptions</b>	The correct directory was selected that contains both the WMP database and WMP Prefetch files

(UC Description:Create GUI and Close WMP Extractor)

<b>Use Case</b>	View report produced by WMP Extractor
<b>Summary</b>	Examiner views report created by WMP Extractor
<b>Actor</b>	User
<b>Trigger</b>	User opens the report using software such as Excel
<b>Primary Scenario</b>	<ol style="list-style-type: none"> <li>1. Examiner opens reports and views media entries extracted from the WMP database and WMP Prefetch files</li> </ol>
<b>Alternative Scenario</b>	<ol style="list-style-type: none"> <li>1. Examiner opens reports but no media entries have been extracted</li> </ol>
<b>Exceptional Scenario</b>	<ol style="list-style-type: none"> <li>1. WMP Extractor is unable to run due to no JVM installed upon the computer</li> </ol>
<b>Pre-Conditions</b>	WMDbExtractorandPrefetchSearch extract entries from the WMP database and WMP Prefetch files and store them in the report
<b>Post-Conditions</b>	Report produced helps prioritise case
<b>Assumptions</b>	WMP Extractor extracts media entries from the WMP database and WMP Prefetch files

**(UC Description:** View report produced by WMP Extractor)



<b>Use Case</b>	Initiate WMP Extractor
<b>Summary</b>	Examiner runs WMP Extractor either in a live or static analysis situation
<b>Actor</b>	User
<b>Trigger</b>	Examiner runs WMPEXtractor
<b>Primary Scenario</b>	1. WMPEXtractor loads and GUI is presented to the examiner
<b>Alternative Scenario</b>	The JVM is not installed and WMP Extractor cannot run
<b>Exceptional Scenario</b>	JVM installed to be able to run WMP Extractor
<b>Pre-Conditions</b>	Computer has the JVM installed
<b>Post-Conditions</b>	Examiner selects directory using the GUI that contains the WMP database and WMP Prefetch files
<b>Assumptions</b>	JVM installed upon the computer

(UC Description: Initiate WMP Extractor)

### 7.3.Pseudo Code

#### 7.3.1.WMPEXtractor

1. Initiate GraphicalUserInterface

#### 7.3.2.GraphicalUserInterface

1. Get forensic examiners name
  1. Error check
2. Get case number
  1. Error check
3. Display Graphical User Interface
4. Allow forensic examiner to run WMP Extractor
  1. Run WMP Extractor using default location
  2. Run WMP Extractor using selected file path
2. Send file path to WMDBFileSignatures

### 7.3.3.WMDNFileSignatures

1. Receive file path from `GraphicalUserInterface`
2. Send file signatures to `WMDBExtractor`
  1. Wait for `WMDBExtractor` to finish searching for file entries
3. Repeat step two until no more file signatures to search for
4. `InitiatePrefetchSignatures`

### 7.3.4.WMDBExtractor

1. Receive file signature from `WMDBFileSignatures`
2. Receive file directory from `GraphicalUserInterface`
3. Check that `CurrentDatabase.wmdbexists` within received directory
  1. If *CurrentDatabase\_372.wmdb* does not exist inform forensic examiner, run `PrefetchExtensions`
4. Read in file using Hex class looking for file entry received from `PrefetchExtrnsions`
5. Extract entries in Hex, pass to `HexConverter`
6. Repeat steps 4 and 5 until every byte has been read in

### 7.3.5.PrefetchExtensions

1. Receive file signature from `PrefetchExtensions`
2. Receive file path from `PrefetchExtensions`
3. Check that WMP Prefetch files exists
  1. If *WMPPLAYER.EXE-xxxxxxxxpf* does not exist inform forensic examiner, finish analysis
4. Read in file using Hex class looking for file entry received from `PrefetchExtensions`
5. Extract entries in Hex, pass to `HexConverter`
6. Check for other WMP Prefetch files

7. Repeat steps 4 and 6 until every byte has been read in

#### 7.3.6.PrefetchSearch

1. Receive file path from WMDBExtractor
2. Send file extension to PrefetchSearch
  - 2.1 Wait for PrefetchSearch to finish searching for entries
3. Repeat step two until no more file extensions to search for
4. Inform forensic examiner analysis has finished

#### 7.3.7.HexConverter

1. Receive hex from WMDBExteactor and PrefetchSearch
2. Convert hex to ASCII
3. Send ASCII to OutputToFile

#### 7.3.8.DateAndTime

1. Receive hex string from WMDBExtractor
  2. Convert hex into date and time
  3. Send date and time to OutputToFile
- 
1. Get current system date and time
  2. Return date and time

#### 7.3.9.FileSize

1. Receive hex string from WMDBExtractor
2. Convert hex string into decimal number
3. Convert decimal number into a string
4. Send string to OutputToFile

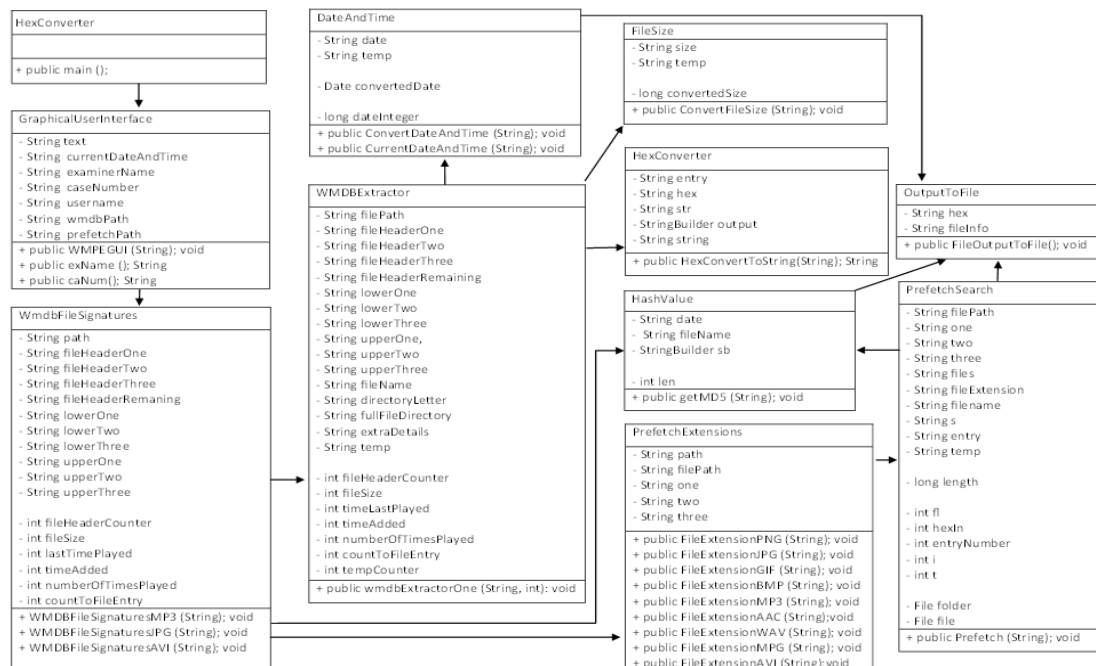
#### 7.3.10.OutputToFile

1. Receive data from  
GraphicalUserInterface,WMDBExtractor,PrefetchSearch,HexConverter,FileSizeandDateAndTime
2. Output information to Tabbed Separated Value (TSV)

#### 7.3.11.HashValue

1. Receive full file path and file name from  
WMDBExtractorandrefetchSearch
2. Calculate MD5 hash value
3. Output MD5 hash value to OutputToFile

## 7.4. Class Diagram



## 7.5.Design Implementation

### 7.5.1.WMPExtractor– class

The only purpose of the main class (WMPExtractor) is to create a new instance of the GraphicalUserInterface and run it. While it would be possible to integrate the main class within the GraphicalUserInterface class, it was advised during *Programming One* (CG0047) that separating the *main class* from the GraphicalUserInterface class conforms to industry standards best practices for better code maintenance.

### 7.5.2.GraphicalUserInterface – class

While this project is a proof of concept, it has been decided within *Software Requirements and Justification* (2.11) to create a Graphical User Interface (GUI) to allow the forensic examiner to perform an analysis with more ease. The GUI will allow the forensic examiner to select the option to perform the analysis, without a GUI the forensic examiner would need to input commands to control WMP Extractor. Using a GUI negates the need for the forensic examiner to remember the commands to control WMP Extractor; a GUI also allows the control of the order in which operations are conducted by only allowing certain options to be selected depending on the stage of analysis.

As described within *Design of the Graphical User Interface (GUI)* (section 6.3) the GUI is going to be as simple as possible as it does not to perform many functions.

- Input Validation

As described within *Design of the Graphical User Interface (GUI)* (section 6.3) input dialog boxes will be used to collect the forensic examiner name and case number. To ensure that the forensic examiner always inputs their name and case number they are required to input the before they progress to the next stage of the analysis. The forensic examiners name should only contain letters and should not be able to be left blank.

```
if(examinerName.isEmpty() || !examinerName.matches("[A-Za-z]*"))
```

The first part of the if statement `examinerName.isEmpty()` checks that the examiners name is not blank, by using `isEmpty()` the examiner is not able to put a space to try and avoid entering a name.

The second part `!examinerName.matches("[A-Za-z]*")` checks if the forensic examiners name does not match any letter a through to z, in either upper or lower case. If the forensic examiners name does not match this criterion the name contains either numbers or special characters. The use of the or `||` statement between both checks ensures that the forensic examiners name must match both criteria or it is rejected and the examiner is presented with the input dialog box. There is no minimal amount of characters an examiner must enter; this allows an examiner to enter in a single letter to identify them if they do not wish their name to be present within the

report. If the examiner was to be required to enter a minimal amount of characters for their name another or || statement could be introduced. The or || statement would check to see if the amount of characters is less than those required by using the notation < x; with x denoting the amount of characters required.

```
if(examinerName.isEmpty() || !examinerName.matches("[A-Za-z]*")) || examinerName < x)
```

Like the forensic examiners name caseNumber.isEmpty() is used to ensure that the case number is not blank. The case number can only contain numbers unlike the forensic examiners name which can contain only letters. By using !

caseNumber.matches("[0-9]\*") ensures that the case number only contains numbers by checking if any of the input does not match number zero through to nine. Both of these conditions are used together by using an or || statement, this insures that both conditions must be met prior to progressing to the next stage of WMP Extractor.

```
if(caseNumber.isEmpty() || !caseNumber.matches("[0-9]*"))
```

Prior to using isEmpty(), examinerName == null was used. While using examinerName == null prevents the forensic examiner from submitting a blank name; even with a space, it prevents the forensic examiner from clicking cancel if they wish to stop WMP Extractor running.

- Main Graphical User Interface (GUI)

To create the design shown within *Design of the Graphical User Interface (GUI)* (section 6.3) three main components were used to build up WMP Extractors GUI.

1. JPanel

The JPanel allows other components such as the JMenuBar and JTextArea to be added to it. The JPanel works similar to blank canvas, components can be added to it to build up the desired GUI.

2. JMenuBar

The JMenuBar works in the same way that the menu bar works in EnCase and other Windows® based software. As described within *Design of the Graphical User Interface (GUI)* the JMenuBar will be placed at the top of WMP Extractors GUI

3. JTextArea

The JTextArea will allow the examiner to be provided with feedback about the state in which WMP Extractor is in, an example is once WMP Extractor is loaded; instructions on what to do next are printed to the JTextArea.

### 7.5.3.WMDBFileSignatures- class

The purpose of `WMDBFileSignatures` is to send the file path, file signature used to find an entry within the WMP database, details of how many bytes the file entry information is within the WMP database file and the file extension to `WMDBExtractor` for each media type i.e. MP3, JPG, AVI. As described within *Overview of CurrentDatabase\_372.wmdb* (2.7) depending on the type of media entry be searched for, depends on what information is available within the WMP database and where it is located after the file signature. Information from `WMDBFileSignatures` is passed into `WMDBFileSignatures` using both *Strings* and *ints*. Each media type has a different file signature within the WMP database.

### 7.5.4.WMDBExtractor- class

The purpose of `WMDBExtractor` is to search for entries of the given media type within the WMP database file, and then extract that information. To enable the reuse of code within `WMP Extractor` one class was used to search the WMP database for all media types. To enable `WMDBExtractor` to find a file entry each byte of the WMP database is read in, and the first byte of the known file signature is looked for using the file signature received from `WMDBFileSignatures`. An example is that the file signature of an audio entry is F2 72 53 05 55. Once the first byte within the file header is found the next byte is checked to see if it matches the second byte within the known file signature i.e. 72. This process is repeated until the entire file signature is found within WMP database thus confirming an entry or, if the byte after the first, second, third or fourth does not match the next byte in the file signature then the search is started again.

Once a file entry is confirmed within the WMP database by the finding of entire file signature in order, `WMDBFileSignatures` scans through a predefined amount of bytes determined by the media entry type and passed in from `WMDBFileSignatures`. Depending on the type of information being extracted `WMDBExtractor` extracts a predefined amount of bytes which is then passed to the respective class. If `WMDBExtractor` is extracting the number of times a media file has been viewed it will extract one byte and pass this to the `HexConverter` class to be converted from Hex into a number. Or if `WMDBExtractor` is extracting the date and time it would extract eight bytes from the WMP database and pass them to the `DataAndTime` class.

When a user names a file such as an audio MP3 file they are not only able to change the name but the extension also. `WMP Extractor` was originally designed to look for lower case and upper case file extensions within the WMP Database, e.g. Northumbria.mp3 or Northumbria.MP3. With the users' ability to name extension using both lower and upper case `WMP Extractor` was adapted to combat this issue. While looking for the extension within the WMP Database both the lower and upper case file extensions in any combination are searched for, e.g. mp3, MP3, Mp3, mP3. The use of an `or ||` statement was used to allow `WMP Extractor` to find both lower and upper case extensions.



```
(lowerOne.equalsIgnoreCase(s) ||
upperOne.equalsIgnoreCase(s))
```

#### 7.5.5.PrefetchExtensions – class

As mentioned within *Overview of CurrentDatabase\_372.wmdb* (2.7) all entries within the *WMPLAYER.EXE-xxxxxxx.pf* all start with `\.D.E.V.I.C.E.\`. With all entries starting with `\.D.E.V.I.C.E.\` and all entries being in upper case, the `PrefetchSearch` class only needs the file extension of the type of entry it is searching for, as the beginning of the file can be hard coded within `PrefetchSearch`. The purpose of `PrefetchExtensions` within `WMP Extractor` is to pass the extension of a file entry to `PrefetchSearch`, once `PrefetchSearch` has looked for that file extension within a `Prefetch` file `PrefetchExtensions` passes in the next file extension to be searched for.

While it would be possible to have a separate class similar to `PrefetchSearch` dedicated looking for one type of file extension. It would also be possible to have one class with multiple methods, with each method looking for a particular file extension; this would be an extremely inefficient way of implementing the `Prefetch` search within `WMP Extractor`. There are approximately one hundred and fifty lines of code that make up `PrefetchSearch`, there are nine different extensions types of within `PrefetchExtensions` which would result in approximately one thousand three hundred and fifty lines of code if each extension had its own class or method. By passing in extensions from `PrefetchExtensions` to `PrefetchSearch` the amount of lines of code needed have been reduced to approximately two hundred and seventy, by using less code overall helps when looking problems within `WMP Extractor` as less code needs to be checked. Using this method has the advantage that extra extensions can be added at a later date by adding only nine lines of code compared to one hundred and fifty.

#### 7.5.6.HexConverter – class

The sole purpose of the `HexConverter` class is to receive strings of hex from both the `WMDBExtractor` and `PrefetchSearch` classes, convert them to strings of ASCII and output the results to `OutputToFile`.

After reviewing several books including *Introduction to Java* by Liang the author was unable to find any information on how to convert strings of Hex to strings of ASCII efficiently (Liang, Y.D. b 2008). Several examples were found that required the use of arrays to convert hex to ASCII. The arrays were populated with known hex characters and their known equivalent in ASCII, each hex character would then be checked against the hex received from both the `WMDBExtractor` and `PrefetchSearch` classes. Converting hex to ASCII using arrays would become very inefficient as the arrays would need to contain every possible ASCII character and their equivalent in hex.

To improve the efficiency of `WMP Extractor` a solution to this problem was found upon a programming forum `Stack Overflow` (`Stack Overflow`, 2011). First a `StringBuilder` is constructed so the converted hex is stored within the variable

*output*. A `StringBuilder` is used to store the converted hex as it allows for easy manipulation such as appending hex onto the end.

```
StringBuilder output = new StringBuilder();
```

The string of hex received from both `WMDBExtractor` and `PrefetchSearch` classes and passed one byte at a time through a loop and converted into ASCII, the converted ASCII is the appended to the end of the *output* variable.

```
for (int i = 0; i < hex.length(); i+=2){  
    String str = hex.substring(i, i+2);  
    output.append((char) Integer.parseInt(str, 16));  
}
```

The converted ASCII is then passed to `OutputToFile` class. Because the `OutputToFile` class is only able to receive Strings from other classes the `StringBuilder` variable *output* is cast as a `String`.

```
String string = output.toString();
```

#### 7.5.7. `DateAndTime` – class

The `DateAndTime` class has two functions. The first function in the method `convertDateAndTime` is to convert Windows® time to Unix time and output it to the `OutputToFile` class.

Windows® uses Coordinated Universal Time (UTC) to record dates and times within the WMP database. UTC time is calculated using the number of one hundred nanoseconds since 1<sup>st</sup> January 1601. Within the WMP database dates and times are stored within eight bytes of information. The eight byte hex value of 70 45 61 36 33 C3 BD 1C equates to 06/03/2011 19:18:07pm. While the dates within the WMP database are recorded using UTC, Java is only able to read dates using the Unix time. The Unix time is calculated by the number seconds passed since 1<sup>st</sup> January 1970.

For the `DateAndTime` class to convert the 8 bytes of hex received from `WMDBExtractor` it must first convert the hex into decimal. The eight bytes of hex received are stored within the variable *temp*.

```
String temp = date;
```

The eight hex bytes stored within the variable *temp* are then parsed into a 64bit long decimal. The number is stored within `dateNumber` as a long as an integer is unable to store such large numbers.

```
long dateNumber = Long.parseLong(temp, 16);
```

As mentioned above Java uses the Unix time and the dates starting from the 1<sup>st</sup> January 1970, whereas the WMP database are stored using Windows® UTC starting from 1<sup>st</sup> January 1601. To enable Java to work out the time stored within the WMP database the time must first be converted using the *Date* class. The 64bit long decimal

number stored within `dateNumber` represents the number of 100 nano seconds passed since 1<sup>st</sup> January 1601. This number is taken away from *1164447360000000000* which represent the number of 100 nano seconds between the 1<sup>st</sup> January 1601 and 1<sup>st</sup> January 1970. The number remaining is the number of 100 nano seconds since 1<sup>st</sup> January 1970. The remaining number is divided by *10000* which represents the number of seconds since 1<sup>st</sup> January 1970. The *Date* class then converts this into readable date.

```
Date convertedDate = new Date((dateNumber -
1164447360000000000L) / 10000);
```

The `OutputToFile` class is only able to receive `Strings` from other classes, to enable it to be able to receive the date and time it first needs to be converted into `String`. Java converts the date which is stored as an object using `String.valueOf` and puts it into a `String` variable *cd*.

```
String cd = String.valueOf(convertedDate);
```

The second function of `DateAndTime` in the method `currentDateAndTime` is to get the current date and time in the format `yyyy-MM-dd HH-mm-ss` from the computer that `WMP Extractor` is running on. The date and time is then returned as a `String` to be presented to the forensic examiner on the GUI and printed out to the report.

#### 7.5.8.FileSize - class

File sizes within the WMP database are stored using either three or four bytes depending on the type of media entry. `WMDbExtractor` extracts either three or four bytes dependant on the media type and passes it to `FileSize`.

To convert the three or four bytes of Hex into the file size it is parsed into a *long* which converts the Hex into a 32 decimal number.

```
long convertedSize = Long.parseLong(temp, 16);
```

The resulting *long* variable is converted into a *String* to enable to be passed to `OutputToFile` to be stored within the report.

```
String cs = String.valueOf(convertedSize);
```

#### 7.5.9.OutputToFile – class

The purpose of `OutputToFile` is to output the information extracted from the WMP database and Prefetch files into a report. Originally the report was going to be output to a text file. By using a text file to store the extracted data from the WMP database and the Prefetch files, it would enable the examiner to open it with almost any text editing software. Text files within Windows® and other operating system do not contain any formatting, without formatting information within the report may become unaligned and difficult to read.

It was decided that the data extracted from the WMP and Prefetch files should be stored within an Excel spread-sheet, this would allow the data to be stored neatly controlled fashion. Using an Excel spread-sheet would allow each column to hold particular information about an entry, using a spread-sheet allows the forensic examiner to order columns such as arranging by the most viewed file. The below screen shot shows how information is stored, columns are used for different types of information relating to a particular entry and rows for individual entries.

	A	B	C	D	E	F	G	H	I	J
1	Examiner	James	Case Number	1	Report Created	2011-03-09 19:10:59				
2										
3		File Size (bytes)	Times Added	Times Last Viewed	Times Viewed	File Name	Directory Letter	Full Directory Path	Extra Details	
4										

(Figure 25 – shows the format of the report when opened with Microsoft® Excel )

There are two different file types when saving to an Excel spread-sheet, these are a Comma-Separated Value (CSV) file and a Tab-Separated Value (TSV) file. The former of the two uses a comma to separate data into different cells whereas the later uses a tab. It has been chosen that WMP Extractor should produce a report using the TSV file. TSV file was chosen as some entries within the WMP database and Prefetch files contains commas, because of this when data is written to a CSV file data can become fragmented over several cells when intended for one cell. While it would be possible to remove the commas prior to writing the information to the report avoiding data becoming fragmented, removing the commas violates the first Association of Chief Police Officers (ACPO) guideline “No action taken by law enforcement agencies or their agents should change data held on a computer or storage media which may subsequently be relied upon in court.”.

#### 7.5.10.HashValue – class

Evidential integrity must be maintained during the extraction of information from both the WMP database and WMP Prefetch files. To do this WMP Extractor will take the hash MD5 hash value of any file examined both prior to and after the extraction is complete. By comparing values to see if they match evidential integrity can be proven as any changes to the WMP database or WMP Prefetch would result in changes to the MD5 hash value.

Java has the ability to take both the MD5 and SHA1 hash values using a built in class. Using the `java.security.MessageDigest` class, Message Digest are “secure one-way hash functions” (Oracle, 2010).

The complete file path and file name are passed from WMPDBExtractor and PrefetchSearch, an input stream is then created to allow HashValue to link to the file

```
InputStream in = new FileInputStream(fileName);
```

A new instance of Message Digest is created using the MD5 algorithm.

```
MessageDigest md = MessageDigest.getInstance("MD5");
```

The file is read in using a buffer input stream.

```
byte[] buf = new byte[1024];
int len;

while ((len = in.read(buf)) != -1) {
    md.update(buf, 0, len);
}
in.close();
```

Once the entire file is stored within the buffer the hash value is calculated using the Message Digest MD5 hash algorithm.

```
byte[] bytes = md.digest();
```

## 7.6. Testing

### 7.6.1. Windows Media Player Database Testing

Type of Information	File Information Entered	File Information Returned
File size in bytes	Blank database	No results returned
Date and time added to WMP	Blank database	No results returned
Date and time audio file was last played	Blank database	No results returned
Number of views	Blank database	No results returned
File name	Blank database	No results returned
Full file path and name of file	Blank database	No results returned
MD5 Hash	180FF61FA14EFD4D09D DA14040DAA41E	180FF61FA14EFD4D09D DA14040DAA41E

#### 7.6.1.1. Audio

Type of Information	File Information Entered	File Information Returned
File size in bytes	8,414,449	8,414,449
Date and time added to WMP	29/02/2011 16:08	29/02/2011 16:08
Date and time audio file was last played	29/02/2011 16:08	29/02/2011 16:08
Number of views	1	1
File name	Kalimba.mp3	Kalimba.mp3
Full file path and name of file	C: \Users\James\Pictures\Sample Music\ Kalimba.mp3	C: \Users\James\Pictures\Sample Music\ Kalimba.mp3
MD5 Hash	21324B26DAC5D2E113B C7E252F82FA78	21324B26DAC5D2E113B C7E252F82FA78

#### 7.6.1.2. Image

Type of Information	File Information Entered	File Information Returned
File size in bytes	845,941	845,941
Date and time added to WMP	29/02/2011 17:22	29/02/2011 17:22
File name	Desert.jpg	Desert.jpg
Full file path and name of file	C: \Users\James\Pictures\Sample Pictures\Desert.jpg	C: \Users\James\Pictures\Sample Pictures\Desert.jpg
MD5 Hash	ABEAB825EBF98A1EFA 64669E30E0C132	ABEAB825EBF98A1EFA 64669E30E0C132



## 7.6.1.3.Video

Type of Information	File Information Entered	File Information Returned
File size in bytes	367,073,280	367,073,280
Date and time added to WMP	29/02/2011 17:25	29/02/2011 17:25
Date and time video file was last played	29/02/2011 17:25	29/02/2011 17:25
Number of views	0	0
File name	Northumbria.avi	Northumbria.avi
Full file path and name of file	C:\Users\James\Sample Video\Northumbria.avi	C:\Users\James\Sample Video\Northumbria.avi
MD5 Hash	7BECDB9389B73EE9A6 BBB94ACC7E53E3	7BECDB9389B73EE9A6 BBB94ACC7E53E3

Type of Information	File Information Entered	File Information Returned
File size in bytes	367,073,280	367,073,280
Date and time added to WMP	29/02/2011 17:29	29/02/2011 17:29
Date and time video file was last played	29/02/2011 17:29	29/02/2011 17:29
Number of views	1	1
File name	Northumbria.avi	Northumbria.avi
Full file path and name of file	C:\Users\James\Sample Video\Northumbria.avi	C:\Users\James\Sample Video\Northumbria.avi
MD5 Hash	D368B0D9A687475BFE8 1E19E8FC709CC	D368B0D9A687475BFE8 1E19E8FC709CC

## 7.6.2.Windows Media Player Prefetch Testing

JPG

Data within Prefetch file	Data extracted by WMP Extractor
\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.0.1.J.P.G\	\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.0.1.J.P.G\
<b>MD5 Hash value before analysis</b>	<b>MD5 Hash value after analysis</b>
E1E5E9DA539D3401787AD59109B18A 3A	E1E5E9DA539D3401787AD59109B18A 3A



## GIF

Data within Prefetch file	Data extracted by WMP Extractor
\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.G.I.F\	\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.G.I.F\
MD5 Hash value before analysis	MD5 Hash value after analysis
C33FF7983034F7791DFCEB450BB6B8 A2	C33FF7983034F7791DFCEB450BB6B8 A2

## BMP

Data within Prefetch file	Data extracted by WMP Extractor
\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.B.M.P\	\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.B.M.P\
MD5 Hash value before analysis	MD5 Hash value after analysis
A33F306246CA245E10E2CA275B42666 E	A33F306246CA245E10E2CA275B42666 E

## MP3

Data within Prefetch file	Data extracted by WMP Extractor
\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.M.P.3\	\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.M.P.3\
MD5 Hash value before analysis	MD5 Hash value after analysis
1B67121A1415C1BE5E65D3C24E1F35 9B	1B67121A1415C1BE5E65D3C24E1F35 9B

## AAC

Data within Prefetch file	Data extracted by WMP Extractor
\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.A.A.C\	\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.A.A.C\
MD5 Hash value before analysis	MD5 Hash value after analysis
696D2B6AD7D7453E30D26BB6E313EE EE	696D2B6AD7D7453E30D26BB6E313EE EE

## WAV

Data within Prefetch file	Data extracted by WMP Extractor
\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.W.A.V\	\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.W.A.V\
MD5 Hash value before analysis	MD5 Hash value after analysis
84DC78217C60F10D52069B931BB4B21 8	84DC78217C60F10D52069B931BB4B21 8

## AVI

Data within Prefetch file	Data extracted by WMP Extractor
\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.A.V.I\	\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.A.V.I\
MD5 Hash value before analysis	MD5 Hash value after analysis
FC5FE6A68405F088F10F38466F84530 5	FC5FE6A68405F088F10F38466F84530 5

## MPG

Data within Prefetch file	Data extracted by WMP Extractor
\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.M.P.G\	\D.E.V.I.C.E. \H.A.R.D.D.I.S.K.V.O.L.U.M.E. 2.\D.R.O.P.B.O.X.\T.E.S.T.S.H.A.R.E.D. \T.E.S.T.O.1.M.P.G\
MD5 Hash value before analysis	MD5 Hash value after analysis
67F48C64AC48EE5D63742F8F2C5B32 B7	67F48C64AC48EE5D63742F8F2C5B32 B7

## 7.6.3. Test Cases

Unique ID	Test 02
Test Description	Examiner name cannot contain numbers
Pre-Conditions	Programming running

Test Data Used	Examiner name entered with numbers
Expected Result	User advised examiner name cannot contain numbers
Actual Result	User advised examiner name cannot contain numbers
Unique ID	Test 03
Test Description	Examiner name cannot contain special characters
Pre-Conditions	Programming running
Test Data Used	Examiner name entered with special characters
Expected Result	User advised examiner name cannot contain special characters
Actual Result	User advised examiner name cannot contain special characters

Unique ID	Test 04
Test Description	Examiner name accepts valid input
Pre-Conditions	Programming running
Test Data Used	Examiner name entered with just letter
Expected Result	Examiner name is accepted
Actual Result	Examiner name accepted

Unique ID	Test 05
Test Description	Case number cannot be left blank
Pre-Conditions	Programming running
Test Data Used	Case number entered blank
Expected Result	User advised case number cannot be left blank
Actual Result	User advised case number cannot be left blank

Unique ID	Test 06
Test Description	Case number cannot contain special characters
Pre-Conditions	Programming running
Test Data Used	Case number/name entered with special characters
Expected Result	User advised case number/name cannot contain special characters
Actual Result	User advised case number/name cannot contain special characters

Unique ID	Test 07
Test Description	Case number cannot contain letters

Pre-Conditions	Programming running
Test Data Used	Case number entered with special letters
Expected Result	User advised case name cannot contain number
Actual Result	User advised case name cannot contain number

Unique ID	Test 08
Test Description	Case number accepts valid input
Pre-Conditions	Programming running
Test Data Used	Case number entered with just number
Expected Result	Case number accepted
Actual Result	Case number accepted

Unique ID	Test 09
Test Description	Default file locations finds files automatically
Pre-Conditions	Examiner name, case number/name and file name entered correctly
Test Data Used	Default location selected
Expected Result	WMP Extractor finds files in default locations
Actual Result	WMP Extractor finds files in default locations

Unique ID	Test 10
Test Description	WMP database file exists but Prefetch does not in default location
Pre-Conditions	Default file location selected
Test Data Used	No Prefetch files exists within default location
Expected Result	User advised no Prefetch files exists
Actual Result	User advised no Prefetch files exists

Unique ID	Test 11
Test Description	User able to select default file directory
Pre-Conditions	Examiner name, case number/name and file name entered correctly
Test Data Used	Default directory selected
Expected Result	WMP Extractor searches for files within default directory
Actual Result	WMP Extractor searches for files within default directory

Unique ID	Test 12
Test Description	User able to select non-default file directory
Pre-Conditions	Examiner name, case number/name and file name entered correctly
Test Data Used	Non-default directory selected
Expected Result	WMP Extractor searches for files within default non-directory
Actual Result	WMP Extractor searches for files within default non-directory

Unique ID	Test 13
Test Description	No files found within default file directory
Pre-Conditions	Default file directory selected
Test Data Used	No files within the default file directory
Expected Result	User advised no files were found within the default file directory
Actual Result	User advised no files were found within the default file directory

Unique ID	Test 14
Test Description	Prefetch file exists but WMP database does not in default file directory
Pre-Conditions	Default file directory selected
Test Data Used	No WMP database file exists within default file directory
Expected Result	User advised WMP Database does not exists, program continues to look for the WMP Prefetch files within the default file directory
Actual Result	User advised WMP Database does not exists, program continues to look for the WMP Prefetch files within the default file directory

Unique ID	Test 15
Test Description	WMP database exists within the default file directory but WMP Prefetch files do not
Pre-Conditions	Non-default file directory selected, WMP database analysed or skipped due to no being present
Test Data Used	No WMP Prefetch files exists within default file directory
Expected Result	User advised WMP Prefetch does not exists within the default file directory
Actual Result	User advised WMP Prefetch does not exists within the default file directory

Unique ID	Test 14
Test Description	No files found within non-default file directory
Pre-Conditions	Non-default file directory selected
Test Data Used	No files within the non-default file directory selected
Expected Result	User advised no files were found within the non-default file directory

Actual Result	User advised no files were found within the non-default file directory
---------------	--

Unique ID	Test 14
Test Description	Prefetch file exists but WMP database does not in non-default file directory
Pre-Conditions	Non-default file directory selected
Test Data Used	No WMP database file exists within non-default file directory
Expected Result	User advised WMP Database does not exists, program continues to look for the WMP Prefetch files within the non-default file directory
Actual Result	User advised WMP Database does not exists, program continues to look for the WMP Prefetch files within the non-default file directory

Unique ID	Test 15
Test Description	WMP database exists within the default file directory but WMP Prefetch files do not
Pre-Conditions	Non-default file directory selected, WMP database analysed or skipped due to no being present
Test Data Used	No WMP Prefetch files exists within default file directory
Expected Result	User advised WMP Prefetch does not exists within the non-default file directory
Actual Result	User advised WMP Prefetch does not exists within the non-default file directory

Unique ID	Test 16
Test Description	Ensure user is able to exit <code>WMP Extractor</code>
Pre-Conditions	Examiner name, case number/name and file name entered correctly
Test Data Used	"Exit" selected from the "File" menu
Expected Result	<code>WMP Extractor</code> exists
Actual Result	<code>WMP Extractor</code> exists

Unique ID	Test 17
Test Description	Ensure user is able to view the help section <code>WMP Extractor</code>
Pre-Conditions	Examiner name, case number/name and file name entered correctly
Test Data Used	"Help" selected from the "File" menu

Expected Result	WMP Extractor displays the help box
Actual Result	WMP Extractordisplays the help box