Portable Storage Forensics: Enhancing the Value of USB Device Analysis and Reporting

MARK SIMMS

A thesis submitted to the graduate faculty of Design and Creative Technologies
AUT University
in partial fulfilment of the
requirements for the degree of
Masters of Forensic Information Technology

2012

School of Computing and Mathematical Sciences

Primary Supervisor: Dr. Brian Cusack

Declaration

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

.....

Signature

Acknowledgements

This thesis was completed at the Faculty of Design and Creative Technologies in the School of Computing and Mathematical Sciences at Auckland University of Technology. I would like to take this opportunity to thank a number of people who assisted me throughout the Research and writing process and who ultimately made it possible for me to complete this thesis.

First and foremost I would like to express my gratitude to my principal supervisor Dr. Brian Cusack for his ongoing guidance and supervision during the current research project. Dr. Brian Cusack along with fellow AUT University lecturers, Dr. David Parry and Krassie Petrova also provided invaluable help in advancing my research and writing skills throughout the program to assist me in pursuing both my academic and professional development goals.

I am extremely grateful to Gareth Jacobs for his technical consultation with some of the more challenging stages of the software coding that were encountered during the prototype design process. Without his advice, expert knowledge of programming and troubleshooting methodologies, the USBForensicReporter© prototype tool would not have been completed to the required professional standard.

I am also indebted to my fellow Master of Forensic Information Technology colleague Jon Pearce for his friendship, support and offering of sound suggestions over the past two years of study. Likewise, I am also very grateful to David Scott and Diana Kassabova for providing insightful critiques and proof-reading of my manuscript. Without their help this thesis would not have got to the final submission stage.

Next, I would like to thank my colleagues across the wider New Zealand digital forensics community who have provided me regular encouragement – your support is greatly appreciated. Finally, I would like to take this opportunity to sincerely thank my wife Joanne and my mother Ann for their continued love and support over the past two and half challenging years. Their underlying faith in me and positive encouragement helped in allowing the goals I am pursuing to be fulfilled.

Abstract

USB based memory storage devices are an easy means of collecting and storing both legitimate and unlawful data. Due to their storage capacity and popularity of use, USB storage devices provide an important source of evidence to both law enforcement and corporate investigations. Digital forensic practitioners are frequently called upon to preserve, analyse, and report USB devices' past connectivity history on Windows® based computer systems.

Existing research and forensic analysis techniques have largely focused on USB artifacts related to the Windows® XP operating system. The release of the Windows® 7 operating system has created new avenues of USB artifact discovery for the digital forensics practitioner. Existing USB and related forensic software tools are plentiful; however, their source code and validation methods are rarely released for public or legal scrutiny. Likewise, there have been no published systematic toolset evaluations of the capabilities and functionality of existing toolsets related to USB device forensics. Consequently practitioners are limited in making the best toolset choices for their analysis needs.

The problem area is USB memory storage device forensics. The purpose of this research was to provide a formal toolset evaluation of existing USB device analysis tools, and to develop a working prototype tool for use in future digital forensic examinations. A set of evaluation criteria was developed in order to identify gaps in existing tools' functionality and reporting standards. The toolset evaluations found each of the tool samples had limitations in forensic functionality or reporting of USB storage devices. A Gap analysis identified three potential areas of improvement in analysis and reporting performance within the sample toolset. These gaps provided sufficient scope for the development of a new software reporting tool in order to add value to and enhance modern USB based forensic recovery techniques.

A working prototype tool named USBForensicReporter© was specifically created as part of the research to support Windows® 7-based USB forensic examinations. The USBForensicReporter© tool provides both accurate and in-depth reporting of USB artifacts.

The tool's design has a unique physical USB device to evidence set comparative analysis method for associating USB storage devices to collected Windows® operating system and registry artifacts. None of evaluated sample tools had this level of comparative analysis whilst employing a single tool interface.

In summary, the software development process was found to add examination value to the discipline of USB based memory device forensics. The developed prototype tool enhanced existing tool functions and providing new comparison analysis and reporting methods for digital forensic practitioners to utilise in the field. Recommendations for future research include releasing a final production version of the prototype software, developing additional tool support for older Windows® operating systems such as Windows® XP, and the anticipated release of the next version, Windows® 8. The toolset benchmarking process also has the potential to be expanded to include a greater range of USB forensic tools for digital forensic practitioners to evaluate.

Publications

Cusack, B., & Simms, M. (2011). Evidential recovery from GPS devices. *Journal of Applied Computing and Information Technology*, 15(1), 1-7. Retrieved from http://www.citrenz.ac.nz/jacit/JACIT1501/2011Cusack_EvidentialRecovery.html

Table of Contents

Declaration ii
Acknowledgements iii
Abstract iv
Publications vi
Table of Contents vii
List of Tables xii
List of Figures xiii
List of Acronyms and Abbreviations xv

Chapter 1: Introduction

1.0 BACKGROUND 1
1.1 PROBLEM AREA, MAIN RESEARCH QUESTION AND HYPOTHESIS STATEMENT 2
1.2 MOTIVATION 4
1.3 STRUCTURE OF THE THESIS 5

Chapter 2: Literature Review

2.0 INTRODUCTION 8
2.1 USB DEVICE OVERVIEW 9
2.1.1 Defining and Examining USB Storage Devices 9
2.1.2 Overview of Key USB Architectural Elements and Specifications 11
2.1.3 USB and Data Loss 15
2.2 THE VALUE OF USB DEVICE EXAMINATIONS 16
2.2.1 Previous USB Forensics Research Based on Windows® XP 18

2.2.2 Tracking USB Activity – The Windows® Registry Way 19
2.2.3 The Next Generations – Windows Vista® and Windows® 7 24
2.3 OTHER OPERATING SYSTEMS AND EXAMINATION RESEARCH .. 24
2.3.1 Linux® Studies 25
2.3.2 Apple® Macintosh® Studies 26
2.3.3 USB Examination Frameworks 27
2.4 PROBLEM AREAS AND ISSUES 28
2.4.1 Windows® XP, Windows Vista® and Windows® 7 Issues 30
2.4.2 USB Examination and Reporting Issues 31
2.4.3 The Emergence of Anti-forensic Techniques 32
2.5 CONCLUSION 33

Chapter 3: Research Methodology

3.0 INTRODUCTION 35
3.1 REVIEW OF PUBLISHED SOFTWARE TOOL STUDIES 36
3.1.1 A Windows® System Restore Software Tool Approach 36
3.1.2 A Windows® Recycle Bin Software Tool Approach 37
3.1.3 A Test Station Software Approach 38
3.1.4 A Mixed-Method Research Approach 39
3.1.5 An Experimental Research Approach 41
3.1.6 Identifying a Preferred Research Methodology 42
3.2 DEVELOPING THE RESEARCH QUESTIONS AND HYPOTHESIS 43
3.2.1 Main Research Question 44
3.2.2 Research Sub-Questions 44
3.2.3 The Proposed Hypothesis 47
3.3 RESEARCH DESIGN 49
3.3.1 A Design Science Approach 49
3.3.2 The Research Methodology 51
3.3.3 The Software Design and Tool Testing Methodology 52

3.4 DATA REQUIREMENTS 58
3.4.1 Data Collection 58
3.4.2 Data Processing and Analysis Methods 59
3.4.3 Data Presentation 60
3.5 LIMITATIONS AND EXPECTED OUTCOMES 61
3.5.1 Limitations of the Sample USB Tool Evaluations 61
3.5.2 Limitations of the New Software Prototype Tool 62
3.5.3 Expected Research Outcomes 62
3.6 CONCLUSION 63

Chapter 4: Research Findings

4.0 INTRODUCTION 64
4.1 MODIFICATIONS TO THE DATA REQUIREMENTS 65
4.1.1 Testing Environment 65
4.1.2 Data Collection 66
4.1.3 Data Processing and Analysis 68
4.2 BENCHMARK TESTING OF THE SAMPLE USB TOOLSET 68
4.2.1 Test Phase One: Configurations and General Setup 69
4.2.2 Test Phase Two: Sample Toolset Evaluations 72
4.2.2.1 Test 1: SanDisk 4 GB Cruzer USB 2.0 Flash Drive 72
4.2.2.2 Test 2: Kingston 4 GB DataTraveler 101 USB 2.0 Flash Drive . 73
4.2.2.3 Test 3: Apacer AH325 4 GB USB 2.0 Flash Drive 73
4.2.2.4 Test 4: Dick Smith 2 GB USB 2.0 Micro Drive 73
4.2.2.5 Test 5: Transcend StoreJet 500 GB USB 2.0 PSD Device 73
4.2.2.6 Test 6: Seagate FreeAgent GoFlex 500 GB USB 2.0 PSD – S1 . 74
4.2.2.7 Test 7: Seagate FreeAgent GoFlex 500 GB USB 2.0 PSD – S2 . 74
4.2.2.8 Test 8: Seagate FreeAgent GoFlex 500 GB USB 3.0 PSD – S3 . 75
4.2.3 Test Phase Three: Sample Toolset Evaluation Overview 75

4.3 ANALYSIS OF THE OVERALL TOOLSET PERFORMANCE 77
4.3.1 Sample Toolset Evaluation Gap Analysis: Phase One 77
4.3.2 Sample Toolset Evaluation Gap Analysis: Phase Two 79
4.3.3 Gaps Identified by the Sample Toolset Evaluations 85
4.4 PRESENTATION OF FINDINGS – TOOLSET BENCHMARKING 88
4.5 CONCLUSION 95

Chapter 5: USB Tool Development

5.0 INTRODUCTION 96
5.1 TOOL DEVELOPMENT 97
5.1.1 Tool Design and Alterations 97
5.1.2 Tool Operation 98
5.1.3 Data Reporting 101
5.2 TOOL VALIDATON 105
5.2.1 Previous Research – Registry Structure Discovery 106
5.2.2 Tool Related Registry and Forensic Artifact Validation 107
5.2.3 Data Verification Method 111
5.3 PROTOTYPE FIELD TESTING AND DATA ANALYSIS 113
5.4 PRESENTATION OF FINDINGS: USB TOOL DEVELOPMENT TESTING 116
5.5 CONCLUSION 121

Chapter 6: Discussion

6.0 INTRODUCTION 122
6.1 DISCUSSION OF RESEARCH FINDINGS 122
6.1.1 Sample USB Toolset Evaluations 123
6.1.2 Gap Analysis 129
6.1.3 Developed USB Prototype Tool 130

6.2 RESEARCH LIMITATIONS 134
6.2.1 USB Tool Evaluations 134
6.2.2 USB Storage Device Limitations 135
6.2.3 Software Design Limitations 136
6.3 RESEARCH QUESTIONS AND HYPOTHESIS 138
6.3.1 The Research Sub-Questions 138
6.3.2 The Main Research Question 142
6.3.3 The Main Hypothesis Test 144
6.4 CONCLUSION 145

Chapter 7: Conclusion

7.0 INTRODUCTION 147
7.1 SUMMARY OF FINDINGS 149
7.2 SUMMARY OF RESEARCH QUESTIONS AND HYPOTHESIS 151
7.3 FUTURE RESEARCH 154

References 156

Appendices

Appendix A – Research Definitions 170
Appendix B – Research Journal 171
Appendix C – Toolset Evaluation Dataset Results 179
Appendix D – Prototype Tool Testing Reports 297
Appendix E – USBForensicReporter Verification Details 353

List of Tables

Table 2.1: USB Specification Release Dates 14
Table 2.2: USB Related Windows® Registry and System Locations 23
Table 3.1: Proposed Toolset Evaluation Template for Evaluation Sheets 57
Table 3.2: Data Types and Reference Locations of Forensic Value 60
Table 4.1: Testing and Analysis Software Requirements for Tool Evaluations . 71
Table 4.2: USB Devices Used During the Toolset Evaluations 72
Table 4.3: Sample Toolset Evaluation Results 76
Table 4.4: Toolset Evaluation: Phase One Gap Analysis Matrix for USB 2.0 Supported Devices 81
Table 4.5: Toolset Evaluation: Phase One Gap Analysis Matrix for a USB 3.0 Supported PSD Device 82
Table 4.6: Toolset Analysis and Reporting Functionality: Phase Two Gap Analysis Results 83
Table 5.1: Test Case Output Verification Results 113
Table 5.2: USBForensicReporter© Field Test Results 114
Table 5.3: Sample Toolset and USBForensicReporter© Tool Analysis Summary 116
Table 5.4: Comparison of Processing Times Captured During the Development Field Testing 117
Table 5.5: Prototype Field Testing Comparison Matrix Conducted for USB 2.0 Supported Devices 119
Table 5.6: Prototype Field Testing Comparison Matrix Conducted for a USB 3.0 Supported PSD Device 120
Table 7.1: Research Questions and Answers Summary 152

List of Figures

Figure 2.1: A USB Architecture Overview 12
Figure 2.2: Standardised Device ID Values 13
Figure 2.3: Windows® Registry Root Keys
Obtained by the Windows® Regedit Tool 20
Figure 2.4: General Registry Structure and Relationships 21
Figure 3.1: Research Road Map 48
Figure 3.2: A Design Science USB Framework for the
Current Research Project 50
Figure 3.3: Proposed Software Design Using CLC and RAD Model Elements . 55
Figure 3.4: Proposed USB Flow Chart for the Functioning of
the Prototype Tool 56
Figure 4.1: USBDeviceForensics© Error in a Virtualised
Test Environment Encountered by the Researcher 66
Figure 4.2: Captured USB Device Descriptor Information by USBlyzer© 67
Figure 4.3: Specific Device Descriptor Information of Interest
Captured by USBlyzer© 68
Figure 4.4: Test-PC Computer Hardware Components and System Details 69
Figure 4.5: Test Hard Drive Configuration and Hardware Details 70
Figure 4.6: Phases One and Two Gap Analysis Methodology Applied
for Identifying Gaps in the Sample USB Toolset 78
Figure 4.7: Phase Two Gap Analysis Findings 87
Figure 4.8: USBDeview© Manual Reporting Results 89
Figure 4.9: EnCase® Forensic Manual Reporting Results 90
Figure 4.10: FTK® RegistryViewer® Manual Reporting Results 91
Figure 4.11: Manual USB Toolset Time Taken Comparison Results 92
Figure 4.12: Phase Two Gap Analysis Results for the Sample Toolset 94
Figure 5.1: USB Prototype Tool Interface and Processing Output 100

Figure 5.2: Completed USB Prototype Tool Operation with the Processing Results Highlighted 101
Figure 5.3: HTML Report Extract Containing Case Data 102
Figure 5.4: HTML Report Shows “No Matches” in the Device Alert Status Field 103
Figure 5.5: HTML Report Shows “Alert Device Found” in the Device Alert Status Field 104
Figure 5.6: The Windows® Registry Hive Structure 107
Figure 5.7: Hive Base Block Extract Details 109
Figure 5.8: Hbin Extract Details 110
Figure 5.9: Key Cell Extract Details 110
Figure 5.10: Value Cell Extract Details 111
Figure 6.1: MountedDevices Subkey Variations in USB Device Types 126
Figure 6.2: MBR Disk Signature Correlation Using FTK® Imager Forensic Software 127
Figure 6.3: HTML Report Extract – Notes Section 133

List of Acronyms and Abbreviations

ASCII	American Standard Code for Information Interchange
BIEN	Bluetooth Information Exchange Network
BIOS	Basic Input/ Output System
CD	Compact Disk
CFTT	Computer Forensic Tool Testing
CLC	Classic Life Cycle
CPU	Central Processing Unit
CR	Conditional Requirements
CRC	Cyclic Redundancy Check
CSV	Comma-Separated Values
DFRWS	Digital Forensic Research Workshop
DLL	Dynamic Link Library
DS	Digital Source
DOJ	Department of Justice
DVD	Digital Versatile Disk
EEPROM	Electrically Erasable Programmable Read-Only Memory
eSATA	External Serial Advanced Technology Attachment
EWF	Expert Witness File
E01	EnCase Evidence File Format
FAT	File Allocation Table
FTK	Forensic Tool Kit
GB	Gigabyte
GUI	Graphical User Interface
GUID	Globally Unique Identifier
HBin	Hive Bin
HTML	Hypertext Markup Language
ID	Identifier

IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
LSB	Least Significant Byte
MB	Megabyte
MBR	Master Boot Record
MD5	Message Digest 5
MSB	Most Significant Byte
NAND	NAND Flash Memory
NAS	Network Attached Storage
NIST	National Institute of Standards and Technology
NOR	NOR Flash Memory
NTFS	New Technology File System
OS	Operating System
PCIe	Peripheral Component Interconnect Express
PSD	Portable Storage Device
RAD	Rapid Application Development
RTF	Rich Text Format
SAN	Storage Area Network
SATA	Serial Advanced Technology Attachment
SDLC	Software Design Life Cycle
SRG	Security Research Group
SRP	System Restore Point
SWGDE	Scientific Work Group on Digital Evidence
TB	Terabyte
UAC	User Account Control
USB	Universal Serial Bus
UTC	Coordinated Universal Time
UIID	Unique Instance Identifier
VV	Verification and Validation
XML	Extensible Markup Language

Chapter 1

Introduction

1.0 BACKGROUND

The traditional field of computer forensics has evolved over the last decade to encompass a broad range of forensic and investigative activities now known as *digital forensics*. Both terminologies have the same scientific basis and are often used interchangeably when discussing investigations involving digital evidence (Carrier, 2006). According to the Oxford Dictionaries Online (Oxford University Press, 2012) the origins of the word *forensic* are related to the Latin meaning of "in open court, public". A more modern interpretation of the term *forensics* is further given by the same source as "scientific tests or techniques used in connection with the detection of crime".

The Scientific Working Group on Digital Evidence (SWGDE, 2011) defined the term *computer forensics* as being the "scientific examination, analysis, and/or evaluation of digital evidence in legal matters" (p.5). Nance, Hay and Bishop (2009) further defined the term *digital forensics* as involving a "wider variety of digital devices" (p.4) than more traditional networks and computer systems that form part of an investigative process. Similarly, Peisert, Bishop and Marzullo (2008) suggest that *digital forensics* is more related to the "the inclusion of devices other than general-purpose computer systems, such as network devices, cell phones and other devices with embedded systems" (p.105). Therefore in the context of current research, the term *digital forensics* implies the examination and analysis of digital storage devices through the use of scientific methods and techniques to detect the presence of criminal activity and/or wrong doing.

The modern digital forensics practitioner faces a host of challenges when it comes to examining an ever expanding array of information technologies, computer systems, digital storage mediums, and examination case scenarios (Casey, 2011). One such challenge is the forensic analysis of Universal Serial Bus (USB) based memory storage devices. USB storage devices are now more than ever manufactured with smaller form factors, greater storage capacities, and broader functionality of use.

As USB plug-n-play technology has become more ubiquitous, many different data storage types are now being encountered by consumers and technology based professionals. Examples include USB thumb drives, portable hard drive units, digital audio players such as iPods and mp3 players, digital cameras and smart phones.

The universal nature of USB based memory technology means there are greater demands being placed on digital forensic practices to perform a wider range of USB related examinations and data interpretations. USB based memory device forensics involves the forensic examination of both computer systems and USB storage devices in an attempt to gather related information from both sources about past USB related activity. A USB forensic examination can add evidential worth to both criminal and civil investigations by the linking of associated data to computers, user activity and USB storage devices. However, each has its own set of complexities and analytical challengers that will be explored in the following chapters.

1.1. PROBLEM AREA, MAIN RESEARCH QUESTION AND HYPOTHESIS STATEMENT

The research problem has three important aspects that are directly related to both present and future USB forensic examinations. The three aspects are identified as: the lack of standardised and universally accepted USB memory/storage frameworks; the lack of published data validation methods for forensic tools, and the lack of formally published toolset evaluation results to enable practitioners to assess capability, accuracy and appropriateness of use. All three aspects are interlinked and will be explored in the current research to provide answers to the research questions in order to provide solutions for the problem area.

Past research has not identified universally accepted and standardised frameworks that are specifically related to USB based memory device forensic examinations. A well-structured and rigorously proven examination framework that has scientific foundation is more likely to be universally accepted by the judicial bodies and industry collectives than a framework that is ad-hoc in nature or based on incomplete or untested criteria and practises.

Most digital forensic processes will make use of existing or customised frameworks that are based on critical aspects of accepted investigation practice such as Collection and Preservation, Examination and Analysis, Presentation and Reporting phases (DFRWS, 2001). Existing frameworks maybe be too general in nature and might not take into account the detailed and unique requirements of USB based device and Windows® Registry analysis. Evaluation and software design frameworks will be developed in the current research to address this component of the problem area.

Software developers and organisations providing commercial forensic software tools generally do not publicly disclose tool validation methods or results due in part to commercial sensitivities around market share or intellectual property concerns. According to Sommer (2010) and Erbacher (2010) any data output that has been produced by software must be based on scientific fact and validated throughout all phases of the forensic process before it can be produced as forensic evidence in a judicial setting. As validation of processes and data output is critical to the production of credible forensic results, the current research aims to produce tool validation methods and results that are both transparent and replicable. Each phase of the research must adhere to a standard that makes it subject to testing and review by both industry and academic peers.

In order for a digital forensic practitioner to choose the correct tool for an examination, the capability and functionality of the tool must be known or testable. Currently, established tool testing programs such as the Computer Forensic Tool Testing (CFTT) project (NIST, 2011) have tested and validated many different forensic hardware and software tools but are not all-inclusive by any means. Beckett & Slay (2007) state that testing conducted by a third party cannot be solely relied upon for evidential purposes without further testing and validation being completed by the intended user or organisation. In the current research, a tool evaluation and validation framework (based on the NIST CFTT principles) will be tested on a sample set of commonly used forensic tools to assist in developing a new forensic tool prototype.

Each of the three aspects of the problem area has now been explained. In order for the problem area to be explored, evaluation and software design frameworks will be developed to assess a sample set of existing USB tools for overall capability and functionality.

Gap analysis results will allow for improvements and new features to be incorporated into the design of a new prototype tool in order to answer the main research question:

What tool design features improve end-user analysis and reporting of USB forensic artifacts?

The main research question and supporting sub-questions will assist in proving or refuting the main hypothesis statement:

USB digital forensic examinations are improved by enhancing the reporting capability of software tools.

1.2. MOTIVATION

A contextual basis to the research problem, hypothesis statement and accompanying research question were introduced in Section 1.1. Specific research areas will be further developed in Chapters 2 and 3. The rationale behind the researcher's selection of the USB forensics topic is based on two motivational factors: enhancing the established body of knowledge in the arena of digital forensics, and professional development. Both factors are inherently linked by the desire to improve forensic methods and discovery of new tools or techniques for the researcher and other colleagues to use in the field.

The researcher is an industry-based practitioner who is currently employed by a New Zealand based digital forensics laboratory. USB based forensic examinations are conducted on a frequent basis in the researcher's laboratory. Over the past six years, a number of technical issues and limitations have been observed in commercial and freeware forensic software offerings by researchers and the author alike. Each limitation has the potential to make forensic examinations more difficult to conduct, time consuming, and prone to errors on the tool or user's part (Richard & Roussev, 2006; Ayers, 2009 and Erbacher, 2010).

For these reasons there is a strong motivation on the researcher's part to improve the functionality of software tools in order to decrease the overall manpower and time costs associated with USB based forensic examinations for both the employer and the wider forensics community.

Furthermore on a personal level, conducting comprehensive research into a specialised area of the forensic discipline will further advance the researcher's professional development and expertise relating to the chosen subject matter.

Past research has primarily focused on USB device forensics relating to the Windows® XP operating system and the discovery of artifacts derived from the Windows® Registry (Farmer, 2005; Luo, 2007 and Carvey 2009). The Windows® Registry offers a wealth of evidential pickings for the forensic practitioner and investigator alike. Therefore the key aims of this research are to: conduct a formal USB toolset evaluation, and to successfully develop a new USB and Windows® 7 based analysis and reporting tool that provides both comprehensive and reliable forensic output for future evidential use in the field.

In short, the developed tool will be built by a forensic practitioner for forensic practitioners to utilise in their forensic toolkits. The selected software design and research methodologies will also utilise a more scientifically based approach open to other forensic tool projects. The purpose of each research approach is to achieve a prototype design that is forensically sound, useful in operation and offers improvements in functionality over existing forensic software.

1.3. STRUCTURE OF THESIS

The thesis consists of seven chapters. The first four are related to the introduction, literature review, research methodology, and toolset evaluation findings. The final three chapters are related to the prototype tool development, a discussion of the research findings and general conclusion. The overall structure follows a logical progression as described below.

Chapter 1 provides a general background and more in-depth outline to the research subject and thesis content. The three aspects of the problem area are discussed before the main research question and hypothesis statement are stated. The rationale and motivations behind selecting the USB forensics subject are then discussed. The chapter concludes by providing a brief summary for each of the remaining chapters.

Chapter 2 provides a broad review of literature related to the use of USB based memory device forensics. This review builds from an overview of the USB specification and physical device characteristics to a more targeted exploration of existing research.

Areas of existing research include Windows® and non-Windows forensic examinations, frameworks and the emergence of anti-forensic techniques. A number of problem areas and issues are also identified in relation to the lack of Windows® 7-based research and the lack of USB memory/storage frameworks, toolset evaluations and validation methods. The current research will focus on the development of a new forensic tool in order to address some of the gaps found in past research.

Chapter 3 uses five Information Technology (IT) and digital forensic studies that are specifically related to established tool development research to identify what approaches and methodologies different researchers have used in past research to develop a methodology for the current research study. A mixed-method research strategy consisting of quantitative, qualitative, design science and software development elements is adopted. The hypothesis, main research question and supporting research sub-questions are derived from the problem areas identified in Chapter 2 and provide a road map for the research project presented in Figure 3.1. The data requirements for the selected research methodology are then outlined to include collection of data related to the Windows registry and tool output, data processing and analysis, and presentation of the toolset evaluation and prototype test results. Expected limitations and outcomes of the research are discussed in the conclusion of the chapter.

Chapter 4 explains a number of modifications that were made to the proposed data requirements before outlining research results related to the sample toolset evaluations and gap analysis phases. The findings in the chapter deliver a previously unpublished benchmark of USB toolset performance measurement for tools that are commonly used in New Zealand digital forensic laboratories. Potential analysis and reporting gaps in the sample toolset are successfully identified through the use of the gap analysis methodology. The performance gaps provide latitude for new features and improvements to be made in the tool design process of the USBForensicReporter© tool. All this helps and to answer the research questions and sub-questions in order to ultimately test the proposed hypothesis statement.

Chapter 5 discusses the development of the USBForensicReporter© tool, including a number of design modifications and unique enhancements to the tool's operation and reporting abilities.

The general tool operation is explained by utilising screenshots of the tool's Graphical User Interface (GUI) and output extracts from the testing phase of the tool development. Tool validation and data verification results from field testing of the developed prototype tool are also reported along with deconstructed registry key and data values.

Chapter 6 discusses the overall research findings by linking the respective findings from the previous two chapters together. Limitations are identified in relation to the overall scope of the research and any potential impact on the findings. The main research question and supporting sub-questions are answered. The main hypothesis statement is tested against the development of new and enhanced analytical and reporting capabilities of the USBForensicReporter© prototype tool.

Chapter 7 concludes the thesis by providing a summary of the major research findings. Answers to the research question and sub-questions are systematically presented in a table format. A schedule for the implementation of the USBForensicReporter© tool from a working prototype to final production version is outlined.

Appendices are provided at the end of the thesis. A list of definitions specifically related to the research study is provided in Appendix A. Appendices B to E are critical elements of the study. These four appendices support the final research findings by providing comprehensive dataset and bookmarked results collected during the toolset evaluations and development testing of the USBForensicReporter© prototype tool.

Chapter 2

Literature Review

2.0 INTRODUCTION

Information and data storage based technologies have evolved at a rapid rate and consequently there has been a growing demand for associated storage devices to be examined by digital forensic practitioners (Casey, 2010). Both researchers and practitioners confront a host of technical and legal challenges in developing relevant recovery methods to meet the demands of conducting digital forensic examinations. One such challenge is the forensic analysis and reporting of USB based memory storage devices so as to add evidential value to criminal and corporate investigations (Pittman & Shaver, 2010).

Chapter 2 examines the current state of both academic and industry knowledge to form a contextual basis for addressing the problem area introduced in Chapter 1. The review will also assist in identifying USB examination and analysis issues so as to derive a set of research questions and methodologies for further exploration in Chapter 3. The chapter comprises of four sections that are designed to review previous literature from a range of theoretical and field-based forensic research studies.

Section 2.1 presents an overview of USB storage devices before exploring the different architectures, device components and USB specifications that commonly support USB technologies. The section concludes with an explanation of why USB devices have become a popular tool for facilitating criminal activity and data loss. Section 2.2 discusses the evidential value of USB device examinations in both civil and criminal investigations. Windows® based Registry research is then reviewed to determine what USB evidential artifacts can be recovered from common operating systems such as Windows® XP, Windows Vista® and Windows® 7. Section 2.3 introduces non-Windows-based examinations and USB framework research to provide an in-depth overview of examination techniques across the USB forensics spectrum.

Section 2.4 uses the reviewed literature to identify specific problem areas, examination challenges, and anti-forensic tactics that a digital forensics practitioner may face when examining USB devices and Windows-based Registry files for evidential artifacts. Finally, Section 2.5 concludes the chapter by proposing how current USB analysis and reporting practices can be improved for the digital forensics practitioner.

2.1 USB DEVICE OVERVIEW

Terms and acronyms such as USB device, USB thumb drive and Portable Storage Device (PSD) are often used in literature to mutually describe a range of compact and portable flash media storage devices. Such devices usually have a small form factor and offer an inexpensive means of digital storage. Also, USB devices are increasingly able to store larger volumes of data when compared to more traditional storage mediums such as hard drives and recordable media like Compact Disks (CDs) and Digital Versatile Disks (DVDs) (Mokube, 2008). With the increasingly ubiquitous nature of USB devices there are also greater risks for individuals, businesses, government, and academic institutions with respect to data loss and/or malicious activity involving information systems (Alghafli, Jones & Martin, 2010). Having the ability to understand and trace USB activity is vital for the successful investigation and identification of wrong-doing or criminal activity (Luo, 2007). Section 2.1.1 provides a foundation for the investigation of USB device connectivity and data retention. The section discusses memory storage types, key USB components and related specification standards to assist in identifying why such devices have become a tool of choice in criminal activity.

2.1.1. Defining and Examining USB Storage Devices

A standard USB thumb drive has no moving internal parts. Major components include a printed circuit board that contains integrated flash memory chips for data storage and a USB connection interface. The components are then encased in a plastic or metal protective shell (Lee et al. 2008). NAND flash memory in the form of non-volatile Electrical Erasable Programmable Memory (EEPROM) is commonly used in modern USB devices to store data.

Data is stored in sector like storage blocks by means of a small electrical charge therefore preventing data from being lost when the device is unplugged from a computer system (van der Knijff, 2010). NAND Flash has been available since the 1980s and is a popular storage medium because the memory type is cheaper and higher in density by storing more binary values (representing by "0" or "1") in smaller cells than early NOR memory (Toshiba, 2008).

NAND Flash has the potential to retain data for longer periods (up to ten years) than older storage mediums whilst allowing the internal storage blocks (also known as pages) to be erased up to a "one million times over the life of the device" (van der Knijff, 2010, p.389). To prolong flash memory storage, manufacturers have developed mathematical algorithm techniques such as *static* and *wear-levelling* to ensure that not one storage block is overused more than the others across the whole capacity of the device (Breeuwsma, de Jongh, Klaver, van der Knijff, & Roeloffs, 2007).

Several studies have conducted in-depth research into the use and recovery of data from non-volatile NAND flash memory devices such as USB thumb drives. Breeuwsma et al. (2007) conducted an experimental study based on the low-level forensic imaging of 45 USB thumb drives as part of wider research project on embedded flash memory and chip extraction examinations. The researchers analysed non-volatile NAND flash memory block structures to aid in the development of automated data carving scripts. Scripting languages such as Python provided an examination mechanism to improve the chances of recovering deleted data from USB devices, and to allow existing and future forensic tools to interpret the output.

Sansurooah (2009) also conducted a similar in-depth investigation into NAND memory storage. The flash translation layer was studied to identify how the storage medium acted as a go-between for various file systems and hardware devices to give transparent data access to the user. Sansurooah argued that flash memory examinations are complex, the quality of forensic tools is low, and there are no established frameworks or methodologies for such low-level examinations.

Digital forensic practitioners are also experiencing a growth in the examination of USB external hard drives or PSD storage mediums. External USB devices use the same USB technology as USB thumb drives but instead of flash media use a conventional hard drive for data storage.

The external case is made up of two main components, laptop size 2.5 inch or computer size 3.5 inch hard drives, and a printed circuit or controller board that contains one or more mini-USB to Serial Advanced Technology Attachment (SATA) connectors. These particular types of non-volatile USB storage device contain a number of physical moving components in the form of a hard drive motor and a series of rotating platters within the hard drive enclosure. Binary data bits are magnetically written to and read from the storage platters when files are either created or accessed by a user or computer-based process (Mueller, 2011).

No references to academic research and only one limited industry study was found during the literature review that solely related to external USB hard drive examinations. According to Lee (2009), the examination and forensic imaging fundamentals of PSD devices are very similar. This means digital forensic practitioners should be able to forensically image them in the same way as a conventional hard drive by removing the outer casing if required.

2.1.2. Overview of Key USB Architectural Elements and Specifications

Modern USB and PSD memory storage devices support the USB specification. USB devices are known as peripheral devices and can be connected via ports and hubs to a computer system in a star-like Bus topology. According to the USB Implementers Forum (USB-IF), the Bus topology supports up to seven tiers with one host controller supporting up to 127 devices and hubs for data communication and transfer (USB Implementers Forum, 2000). Whilst transparent to the user, connecting a USB device to a computer system requires both the system and the device to interact at various software layers and hardware interfaces. USB operability is ultimately achieved by reading manufacturer information contained on the firmware of the device and loading device drivers to communicate between the host computer and USB device.

Figure 2.1 depicts the USB host controller as being the central communication point for information to flow between host computer and the connected USB device. The host controller initiates and controls data transfers through actively sampling the bus and the connected USB device. When USB devices are connected to a host computer various artifacts or footprints are left in a number of operating system and log files (Farmer, 2007).

Each operating system will record these artifacts in varying degrees and formats so a number of established research and on-going examination methodologies must be employed by the digital forensic practitioner to recover them (Carvey & Altheide, 2005; Sansurooah, 2009).

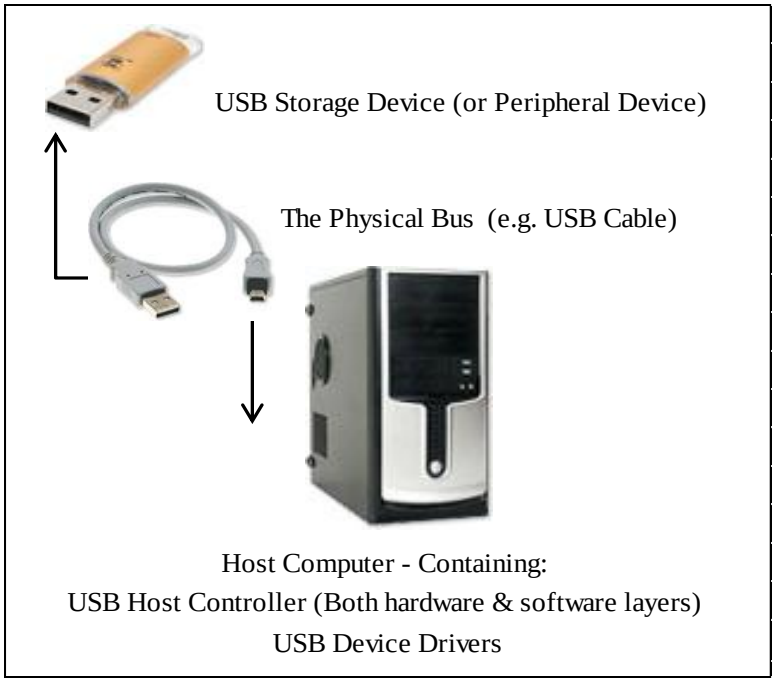


Figure 2.1. A USB Architecture Overview. Adapted from http://www.usblyzer.com/usb-system-architecture-components.htm

In the case of a Windows® operating system, when a new USB storage device is connected to the host computer, certain embedded information contained within the device is retrieved by Windows® in order to locate the appropriate driver package for the device. Microsoft® provides online library resources and documentation for specific type information via the Microsoft® Development Network (MSDN). Microsoft® designates embedded USB device information as *device descriptors* (Microsoft, 2010).

A USB device can only contain a single *device descriptor*, encompassing class information to identify what functionality the device has, thus allowing the appropriate driver to be loaded by the host computer (USB Implementers Forum, 2008). USB devices are divided into different types such as hub, printer and human interface classes. In particular, USB storage devices are identified as a 'mass storage device' class (USB Implementers Forum, 2009).

According to Microsoft® (2010), a unique device identifier named *Device ID* is also created from the *device descriptor* information using the standardised format shown in Figure 2.2.

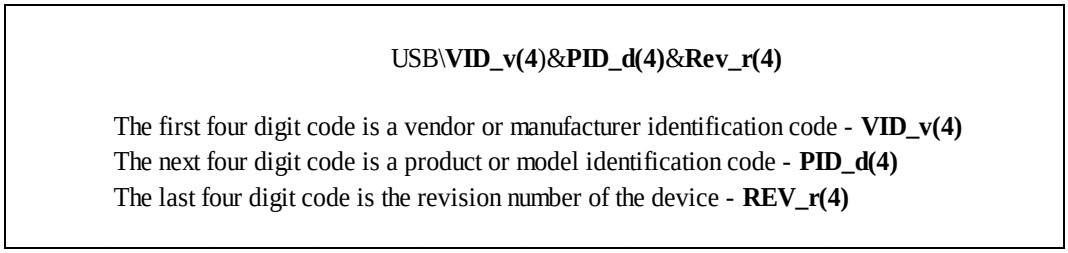


Figure 2.2. Standardised Device ID Values. Adapted from http://msdn.Microsoft.com/en-us/library/ff553356.aspx

A number of research studies have used different sources from Microsoft® produced resource information to explain these device information concepts. Carvey and Altheide (2005) described the *device descriptor* and *Device ID* values in detail and also provided screenshots of a connected USB device to give the reader a visual representation of how *device descriptor* information was recorded. Similarly, Luo (2007) used Carvey and Altheide's earlier Windows® XP Registry study and screenshots as a foundation for his Windows® XP based forensic USB research. Luo expanded on the use of the *device descriptor* information by using both Windows® XP and Linux® operating systems to capture connection activity from the same USB device. The study identified that other non-Windows-based operating systems such as Linux® can also use *device descriptor* information to record USB device information in both log and system files.

Windows-based forensic researchers such as Carvey (2009) suggested that not all manufacturers populate all of the available device descriptor fields in their devices. However, when encountered, the *iManufacturer*, *iProduct* and *iSerialNumber* values found in these fields can provide assistance to digital forensic practitioners during a USB examination. These *device descriptor* values also have the potential to link a physical USB device to a particular computer system via various operating system artifacts and log file entries (Farmer, 2007). Equally so, the values can also cause translation problems for the digital forensics practitioner when a USB storage device does not contain a serial number in the *device descriptor* field.

Section 2.2.2 will further discuss how the Windows operating system creates a Unique Instance Identifier (UIID) value when the manufacturer does not provide an *iSerialNumber*.

The updated Windows forensic research by Carvey (2011) further highlighted other system file areas where *device descriptor* information can be recorded such as the setupapi.log file in Windows® XP and the *setupai.dev.log* in newer Windows Vista® and Windows® 7 operating system versions. Carvey also emphasized the point that to use USB forensic methodologies and examination techniques, a digital forensics practitioner needs to have a technical understanding of the USB technology and operating system storage locations in order to be able to effectively find USB related artifacts that may contain evidential data.

Finally, the USB specification provides a documented and technical understanding of how the different specifications interact with each other and how USB devices are designed to operate. Each of the three main specifications have been developed and released by the non-profit USB Implementers Forum organisation for the benefit of both hardware and software developers, and IT industry based technology specialists. These specification documents are highly technical in nature and sizable with the USB 2.0 specification document coming in at 650 pages (USB Implementers Forum, 2000) and USB 3.0 specification document coming in at 482 pages (USB Implementers Forum, 2008). Table 1 identifies the three USB specification versions that have been released over a twelve year period by the USB Implementers Forum.

Table 2.1

USB Specification Release Dates. (Adapted from USB.Org, 2010).

USB Version	**Release Date**
USB 1.0	January 1996
USB 2.0	27 April 2000
USB 3.0	12 November 2008

Quirk (2011) identified that USB 2.0 supported devices were still very much seen as the "standard" for USB technology in 2010 with upwards of 2.5 billion supported devices being made available to the global marketplace. The uptake of USB 3.0 related technology was also seen to be on the rise by 2011 as Quirk further reporting that close to 80 million USB 3.0 devices was due to be shipped.

In contrast, visits to a number of New Zealand based computer retailers identified that new computer systems were being sold with a varied range of USB port combinations in late 2011 to early 2012. Combinations included a mixture of USB 2.0/3.0 interfaces, legacy USB 2.0 support only, and the newer USB 3.0 support only. Nevertheless, USB 2.0 port support still featured in large numbers amongst a wide variety of new custom-built and branded computer systems.

The advantage of using newer USB 3.0 products is purely for speed purposes with a theoretical throughput of 5.0 Gbps by the addition of a secondary physical bus (also known as SuperSpeed USB) in parallel with the existing USB 2.0 bus (USB Implementers Forum, 2008). From industry feedback there is not likely to be major changes in USB based examinations when USB 3.0 technology is implemented and utilised on a wider scale. Backwards connectivity and compatibility with legacy USB devices will still be maintained through the use of four USB 2.0 contacts that are included in a new USB 3.0 cable connector. The new connector has an additional five USB 3.0 contacts on top of the USB 2.0 contacts that are currently used (Xbit Laboratories, 2010).

2.1.3 USB and Data Loss

Recent industry studies have been conducted to investigate trends in security incidents, computer crime and data loss across both the private and public sectors. As a result, the USB based memory storage device has emerged as a central instrument in facilitating the loss of data from information systems and computer networks.

The risk that USB storage devices pose to both business and government organisations was highlighted in a recent New Zealand computer crime and security survey by the University of Otago Security Research Group (SRG) (Quinn, 2010). Computer security specialists from various New Zealand industrial and tertiary sectors were surveyed on a wide range of current ICT issues using both telephone and email related collection methods. The survey concluded that USB storage devices were a major risk factor in incidents involving the loss of both private and intellectual property data.

Furthermore, USB storage device factored heavily in the spread of malicious software infections across computer systems. The survey results also showed that more than half of the 176 respondents acknowledged no

organisational protection mechanisms were in place to prevent USB related incidents from occurring.

New Zealand based digital forensic experts such as Deloitte Forensics also recognised a growing trend in the use of USB devices to aid in the theft of both private and company related data (Slade, 2011). In an earlier study, Gorge (2005) identified that USB devices are increasingly becoming cheaper to use and more accessible to corporate users. Consequently the form factor of such devices has also made them a popular and easily transportable mechanism to assist in the removal of sensitive data or to import malicious/inappropriate material into corporate networks. As the dangers of unprotected USB usage become more prevalent, businesses are being urged by both government and industry experts to take a more pro-active USB security approach through the use of acceptable usage policies, USB device restrictions and the necessity to perform a detailed USB forensic investigation when a data breach is identified (Privacy Commissioner, 2010).

2.2 THE VALUE OF USB DEVICE EXAMINATIONS

As previously identified in Section 2.1.3, the risk of exposing an individual or organisation to data loss has dramatically increased in recent years as the capacity of USB based memory storage devices has grown and designs have become significantly smaller in size. Since these devices are now more than ever easily lost or stolen they can lead to security breaches resulting in the loss of public trust and professional reputations. Incidents are frequently reported by the media and research organisations to highlight the vulnerability of such devices. Two such examples include the theft of 3.3 million student identity records from a USB portable storage device in Minneapolis, USA (DataLossDB, 2011) and the loss of 6,000 prisoner medical records on a USB memory stick in Lancashire, United Kingdom (BBC, 2009).

Determining what data content can be recovered from USB devices is one of the primary goals of recent forensic research studies. Jones, Valli and Dabibi (2009) conducted a study where-by 43 USB storage devices were collected across the United Kingdom, United Arab Emirates and Australia for forensic examination purposes.

The authors used the same qualitative research methodology as a similar second-hand hard drive study that Jones and Valli had participated between 2005 and 2009. Various USB based memory storage devices were brought from auction houses, online auctions such as eBay and computer fairs before being imaged and analysed with forensic software to determine what data could be recovered and what data loss risk they posed.

The study results indicated that forensic examinations of USB storage devices yielded both current and deleted sensitive corporate or personal data. Jones et al. (2009) further identified that sensitive data could have easily been exploited for more sinister or criminal motives if the USB storage devices had ended up in the hands of criminal organisations. Identifying the link between a physical USB device, computer system and associated user account is the primary goal of USB forensic examinations.

The recovery of USB link data can take the form of device information such as the USB serial number, past USB connection information left in the Windows® Registry of a computer system and file metadata (i.e. information data). Windows® based link file data on the computer system can also assist in identifying that a particular file had been opened from the USB device. Pittman and Shaver (2010) also highlighted that USB device examinations can play an important role in cases involving objectionable material, theft of intellectual property and other computer related crime where associations to users, image and data files have to be made by the digital forensic practitioner to aid criminal and/or civil prosecution proceedings.

Finally, USB devices are now becoming an accepted medium for evidential data recovery in situations where the collection of digital evidence cannot be achieved by attaching a conventional Integrated Drive Electronics (IDE) or Serial Advanced Technology Attachment (SATA) hard drive to a "live" or running computer system. Their use is particularly relevant to active databases or in circumstances involving enterprise or critical web servers that cannot be shut down due to them running mission-critical applications. USB devices are also an important evidence collection container for "live" computer systems running active user encryption applications (i.e. TrueCrypt®, BestCrypt® or PGP® encryption software) or to capture volatile data such as data from Random Access Memory (RAM) and active network traffic in malware investigations.

Furthermore, digital media devices such as proprietary Closed Circuit Television (CCTV) systems may also only allow the practitioner to extract video data by connecting a USB storage device to the unit after previously recorded footage has been reviewed and selected for download (ACPO, 2007).

2.2.1. Previous USB Forensics Research Based on Windows® XP

Over the past six years a limited number of academic and industry studies have investigated the topic of USB forensics with a noticeable trend focusing on Windows® XP operating system based research. Windows® XP was released by Microsoft® in 2001 and has been a popular operating system product for both end users and digital forensic practitioners alike. Although now considered a legacy operating system under the Microsoft® Support Lifecycle, Windows® XP still has more than half (55.27%) of the market share in operating systems when compared to newer Windows® product lines (e.g. Windows Vista® and Windows® 7) and other major operating system platforms such as Apple® Macintosh® and Linux® (Net Applications, 2011). Furthermore, the Windows® XP operating system is a stable and commonly used tool when conducting field trials to allow researchers to better understand how USB device activity is recorded. Carvey and Altheide (2005) used the Windows® XP operating system as an early research model for examining USB thumb drives to identify specific Windows® Registry artifacts pertaining to the past connection of such devices.

Of particular research interest was the identification of recorded USB related activities in a Windows® log file named *setupapi.log*. Under normal installation conditions, the log file is located in *C:\WINDOWS* and is commonly denoted as *%SYSTEMROOT%*. The *setupapi.log* file is a simple plaintext file that records device, service pack and hotfix installations and device/driver changes for that particular installation of the operating system (Microsoft, 2003). Luo (2007) also examined embedded identifiers and footprints of USB devices left in the Registry of Windows® XP operating systems. The author used examination techniques such as screenshots of the Registry Editor and entries in the *setupapi.log* file that are similar to the ones used in Carvey and Altheide (2005) study and are rather primitive compared to current forensic and tool-based methodologies.

When compared, more emphasis in the latter study was placed on tracking a USB device line by line as the device and associated software drivers were installed by the test operating system and recorded in the associated log file. Aside from specific Windows® Registry files that will be further discussed in Section 2.2.2, both of these studies have shown that the *setupapi.log* file can be of great value to the digital forensics practitioner.

The log file provides an additional source of related USB artifacts and offers a timeline of connection history for previously attached USB devices. The *setupapi.log* file can also be used to track other types of media storage devices such as digital cameras and iPods that are increasingly being encountered in cases involving sexual offending and objectionable material examinations (Mokube, 2008). Furthermore, Microsoft® has retained the use of the *setupapi.log* file in newer versions of the Windows® operating system up to Windows® 7, although in a different file format.

2.2.2. Tracking USB Activity – The Windows® Registry Way

Before conducting real-world investigations or USB specific examinations a digital forensics practitioner must have in-depth knowledge and understanding of how the different components of the Windows® Registry work and interact with each other. The Windows® Registry has been at the centre of Microsoft® operating systems since the release of Windows® 3.1 in 1992, although in different forms. Honeycutt (2003) defines the Registry as a "hierarchical database or central repository of configuration data and application settings" (p.3). Furthermore, Honeycutt also identified that the structure of the Windows® Registry is very similar to the hieratical file system and folder management system provided by the Windows® Explorer application on Windows® operating systems. Each operating system version release also brings new entries and subkeys that could contain a wealth of evidential artifacts for the digital forensics practitioner to locate. Specific Windows® based Registry applications such as the Registry Editor tool (Regedit.exe) shown in Figure 2.3 are available to allow a user to display the logical Registry structure, and to search and alter settings on a running computer system.

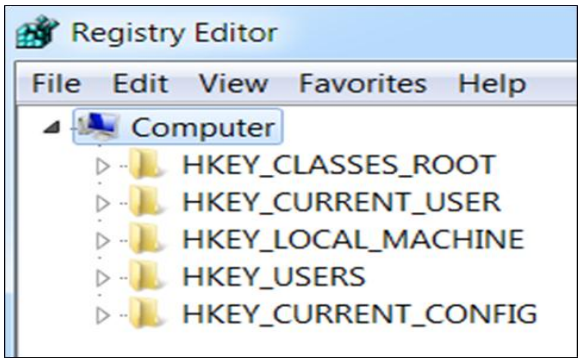


Figure 2.3. Windows® Registry Root Keys Obtained by the Windows® Regedit Tool

The *HKEY_LOCAL_MACHINE* (HKLM) and *HKEY_USERS* (HKU) registry keys shown in Figure 2.3 are the most important of the five root keys. According to Honeycutt (2003) the other subkeys link back to the more important root keys. The key relationship and link relationship can be explained as:

- *HKEY_CURRENT_USER* links to *HKEY_USERS*
- *HKEY_CLASSES_ROOT and HKEY_CURRENT_CONFIG* both link to the *HKEY_LOCAL_MACHINE*

The main root keys are also known as hives and each hive contains various subkeys and data values. Honeycutt (2003) describes keys as being "similar to folders" (p.16) with the same graphical icon being used to display Registry keys and folders in Windows® based operating systems. Figure 2.4 illustrates a representation of how the hives, keys and values interact with each other on a live Windows® Registry. The literature review also determined that the actual main hive names and relationship structure have not changed greatly between the release of Windows® XP in 2001 and the release of Windows® 7 in 2009.

The Windows® Registry provides a rich source of evidential data that can be used by digital forensic practitioners to examine and collect data (Yasin, Cheema & Kausar, 2010). Similar focused studies have also been conducted by academic and industry researchers over the past six years to uncover application and USB device artifacts to benefit Windows® based forensic investigations. The majority of these studies use the Windows® XP operating system as a testing platform to collect Registry and USB artifacts through the use of common forensic software, analysis and reporting techniques.

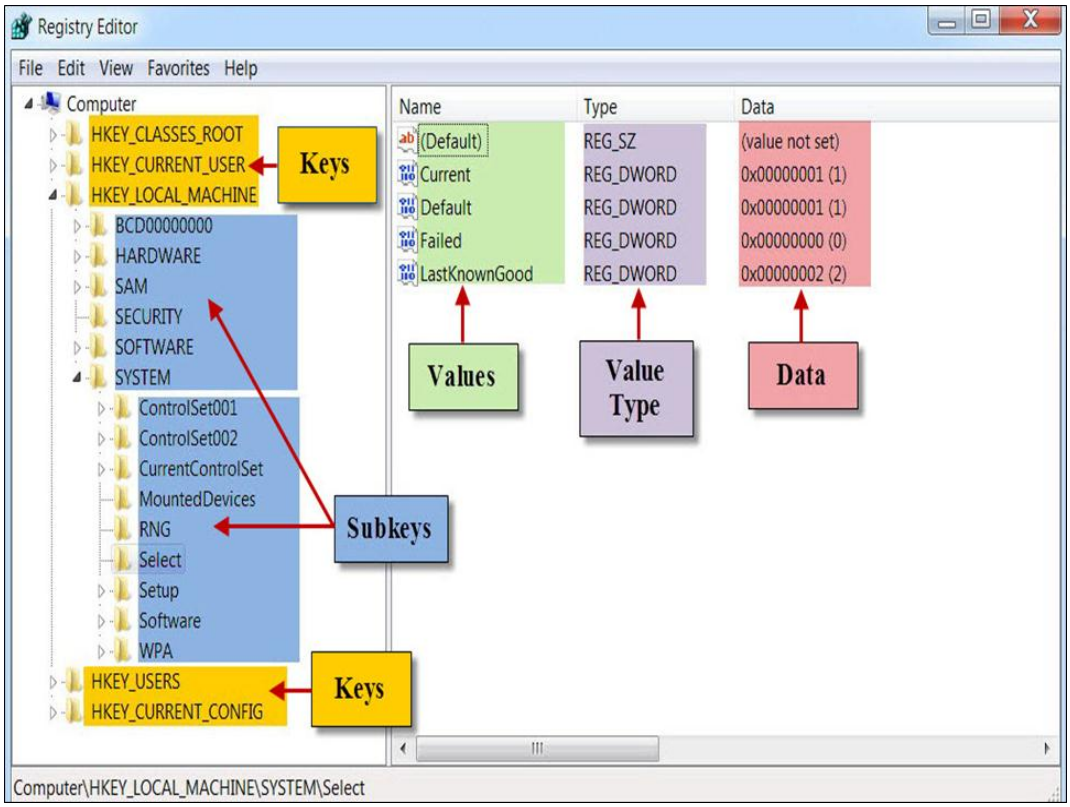


Figure 2.4. General Registry Structure and Relationships. (Adapted from Framer, 2007)

Carvey and Altheide (2005) used the Windows® XP operating system as an early research model to identify specific Windows® Registry artifacts pertaining to USB connection history. The research found that if a USB storage device contained no unique vendor serial number (known as an *iSerialNumber*) then the Windows® Plug and Play Manager would assign a U*nique Instance Identifier* (UIID) value. The *UIID* value is used for identification purposes when multiple devices of the same class type are connected to a system.

These unique identifiers along with the *ParentIdPrefix* value in the *USBSTOR* key can generally assist an examiner in determining a connectivity pattern for the USB device (Alghalfli, Jones & Martin, 2010). However, a limiting factor when using the *UIID* analysis methodology is the fact that the *UIID* values can be misleading for reporting purposes, particularly when the same model of USB device has been connected multiple times and the same drive letter used. In contrast, USB devices with a valid serial number are considered more reliable.

Carvey (2009) expanded on his early research by developing Perl-based automated scripts for the recovery of Windows® Registry artifacts and published the Windows Forensic Analysis DVD Toolkit 2E book in 2009 as an extensive printed resource for digital forensic practitioners. Furthermore, the author identified that the *ParentIdPrefix* value can also be used in the *MountedDevices* key and user specific *NTUSER.DAT* file to determine what "drive letter was assigned to the device and what user account profile had last accessed the device" (Carvey, 2009, p.223). Carvey also identified that the *MountedDevices* key can also create problems for digital forensic practitioners when multiple USB devices have been connected to a Windows® XP operating system and the same drive letter has been mapped to each device.

Identifying and validating the relationships between the various registry keys and associated USB artifacts becomes critical when forensic examinations are conducted in large scale corporate environments. These types of USB examinations are complex, time-consuming and can easy overwhelm a practitioner with considerable processing overheads. Examinations could become quickly bogged down when computer systems across multiple sites are encountered and the practitioner does not make use of automated processing scripts or specialised USB reporting tools.

Luo (2007) also examined the unique USB identifiers and found evidence of previous USB devices having been recorded in the registry of Windows® XP operating systems. The author's research concentrated on the *device ID* and *SYSTEM* hive references (e.g. *USB, USBSTOR* and *MountedDevices* subkeys) that were previously identified by Carvey and Altheide (2005) as forensically informative. Whilst the research replicated previous research studies, the study did however provide a more detailed examination of the *setupapi.log* with screenshots depicting output of an actual USB device being connected to the computer system. Luo further identified that an examiner could compare the *setupapi.log* file entries with the *Last Written* timestamp in the *SYSTEM* hive in an effort to make a determination of the connection history from a particular USB device.

Lee et al. (2008) used traditional examination methods to further investigate bypassing the security functions of controllers on certain types of USB devices that were sold in the Republic of South Korea in 2007.

The authors implemented a software solution known as the USB PassOn© analysis tool and used password sniffing to bypass four controller types that were available at the time in order to read all areas of the USB device being examined. The USB PassOn analysis tool also used a modular approach to display registry and USB related device information from the *SOFTWARE* hive for further reporting purposes.

In summary, specific Windows® based registry artifacts were found across the reviewed literature that proved to be beneficial to USB related forensic investigations. Whilst most of these registry keys are referenced to Windows® XP, a small core of the *SYSTEM* hive keys has remained consistent with Windows Vista® to Windows® 7 including the *USB*, *USBSTOR* and *MountedDevices* subkeys. Table 2.2 provides a list of registry locations that have been identified from the literature review to further assist digital forensic practitioners in finding USB related forensic artifacts.

Table 2.2

USB Related Windows® Registry and System Locations. (Adapted from Windows Forensic Analysis, by Carvey, 2009, pp.206-219, Syngress)

Windows Registry Location	Information Overview
HKLM\SYSTEM\CurrentControlSet\Enum\USB	Vender and Product ID and UIID or ParentIdPrefix value details
HKLM\SYSTEM\CurrentControlSet\Enum\ USBSTOR	Device Class ID and UIID or ParentIdPrefix values. Device make and model information
HKLM\SYSTEM\MountedDevices	UIID values, drive letter and volume GUID mappings
NTUSER.DAT\Software\Microsoft\Windows\ CurrentVersion\Explorer\MountPoints2	Volume GUID and device letter mapping. Used in conjunction with the *MountedDevices* key and GUID to associate past user connection to the USB device
HKLM/SOFTWARE\Microsoft\Windows Portable Devices\Devices	UIID values, FriendlyName details. Record of previous drive letter mapping
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\EMDMgmt (i.e. ReadyBoost Service)	UIID values and LastTested date and time stamp - Windows Vista® and Windows® 7
**System Drive Letter*\Windows\setupapi.log*	1st device connection date and serial number or ParentIdPrefix details - Windows® XP
**System Drive Letter*\Windows\inf\ setupapi.dev.log*	1st device connection date and serial number details - Windows Vista® & Windows® 7
**System Drive Letter*\Document and Settings*UserProfileName* \NTUSER.DAT*	Windows® XP - Location of the user account's NTUSER.DAT File
**System Drive Letter*\Users*UserProfileName*\ NTUSER.DAT*	Windows Vista® & Windows® 7 - Location of a user account's NTUSER.DAT File

2.2.3. The Next Generations – Windows Vista® and Windows® 7

With the release of Windows Vista® in 2007, a number of changes were made to the *setupapi.log* file. The original Windows® XP *setupapi.log* file provided assistance to digital forensic practitioners with the logging of USB device information and has now been replaced by two new files named *setupapi.dev.log* and *setupapi.app.log*. Both of these files have also been relocated from the root of the Windows® system folder to a subfolder named *inf*.

Microsoft® (2011b) identified that the *setupapi.dev.log* file contains both device and driver information as the previous *setupapi.log* file did in Windows® XP. The new *setupapi.app.log* file now contains both legacy and current application logging information. Further research needs to be undertaken to identify if any USB related artifacts can be found within the new file format.

Several studies by Lee (2009) and Carvey (2009, 2011) have also identified that the Windows Vista® and Windows 7® operating systems may assist digital forensic practitioners in collecting additional sources of USB artifacts. The *HKLM\SOFTWARE\Microsoft\Windows Portable Devices\Devices* subkey is one such location source that was identified by both researchers.

An examination of the *Windows Portable Devices* subkey using a forensic copy of a hard drive or on a live system using the Windows® Registry Editor will reveal a wealth of information on previous USB devices. Information relating to USB thumb drives, iPods, USB printers and digital cameras along with the last assigned drive letter can be recorded in this location. Further research in the current study needs to be conducted of the Windows® Registry to identify how the *USB, USBSTOR, MountedDevices* and *Windows Portable Device* subkeys are linked and to see if each of the child or subkeys can contain identical USB device information.

2.3 OTHER OPERATING SYSTEMS AND EXAMINATION RESEARCH

The literature review identified very few USB forensic studies related to non-Windows operating systems. The lack of research balance between operating systems is indicative of the popularity and demand that Windows® based operating systems have with both users and industry professionals alike.

As of February 2011, Net Applications (2011) reported that Windows® operating systems had a total market share of 89.63% when compared to other operating systems. From an examination perspective, industry related practitioners generally deal with Windows® based operating systems on a daily basis and rarely have the opportunity to research or conduct Linux® and Apple® Macintosh® based examinations (Pittman & Shaver, 2010).

Sections 2.3.1 to 2.3.2 provide an underlying understanding of USB examinations across other operating systems. This understanding allows digital forensic practitioners to transfer their USB knowledge from one operating system to another whilst utilising similar examination and recovery principals. A multi-disciplinary approach to USB examinations increases the body of knowledge in a laboratory environment and allows for research opportunities relating to USB forensics to be undertaken as new operating system and USB storage devices are updated and released.

2.3.1. Linux® Studies

USB identifiers such as the vendor name, serial number and device descriptions along with connection records can also be found on non-Windows operating systems. Several studies have identified that log data can be found on a Linux® operating system when USB storage devices have previously been connected to the system. Kemble (2008) tested USB storage devices to identify log artifacts, namely in the *syslog.log* file that could prove valuable to a forensic investigation on Linux® Fedora® 8 operating systems.

Altheide and Casey (2010) also identified that the *syslog.log* file would also contain USB connection information and further added that other Linux-based software applications may have also contained references to file names that are stored on the device itself. The usefulness of log file analysis in a forensic examination was further highlighted by Luo (2007) who used the recorded USB identifiers of a USB device to verify Windows® XP registry connection activity when the same USB storage device was also connected to a Linux® operating system. Luo utilised the *cat* command from the *proc* directory on a Linux® operating system to list device information such as the vendor, serial number and device type.

The USB related output from the *cat* command was then used to validate entries made in Linux® syslog.log file and identical USB information that had previously been found in a Windows® XP Registry.

The approach of using different sources and operating system resources to validate recovered USB artifacts is an important aspect of digital forensic examinations and associated research studies. Pittman and Shaver (2010) highlighted this very point with particular reference to evidential data by arguing that no evidential data from such examinations should be produced in a court of law unless a validation process occurred.

2.3.2. Apple® Macintosh® Studies

When compared to the Windows® Registry, the Apple® Macintosh® (Mac OS® X) operating system does not offer a centralised repository of system and device information for ease of examination. Instead, Mac OS® X relies on different types of system logs and *plist* (property list) files to record historical USB connections. The degree of USB log availability is largely dependent on the particular operating system version. Kokocinski (2010) highlighted USB logging in a study of Macintosh® based forensic analysis and identified that each Mac OS® X version (from 10.2 to 10.5) handles the logging of USB device connections in a different manner. Early versions contained detailed logs whilst later ones contain very few if any logs.

BlackBag™ Technologies (2011) recently identified the *kernel.log* file as a new evidence-related source for USB artifacts in later Apple® operating system versions from version 10.6 onwards. The serial number of past USB devices can be found in the *kernel.log* file by searching for the *USBMSC* identifier in a forensic image copy of the hard drive or on "live" systems via the Disk Utility window to link USB devices to a particular operating system. Several online forensics resources also offer practical advice about the examination of Mac OS® X operating systems. Specific log files such as *system.log* file found in folder location */private/var/log* and the *com.apple.sidebarlists.plist* that is found under /*Users/username/Library/Preferences/* folder location can yield volume, device and connection artifacts that may be of evidential value to a USB related investigation (The Apple Examiner, 2011).

To further assist industry practitioners, Kokocinski (2010) investigated other Mac OS® X system related locations that may contain useful USB artifacts. These locations include the *fsck_hfs.log* file for mounted volume information and the *DiskUtility.log* file for user related device formatting or mounting of disk images. Deleted mount point entries (relating to the connection of Apple® iPods® or USB thumb drives for example) and records from indexing applications such as Spotlight® may also be of assistance in locating USB artifacts on more modern Mac OS® X operating systems.

Mokube (2008) expanded on the theme of USB related Macintosh® forensics by examining the popular Apple® *iPod*® shuffle. The study established that older deleted image files can still be recovered from memory by using standard industry forensic software tools. Recovery is largely due to flash memory devices using an established technique called *wear-levelling* to ensure that no one data sector is overused more than the others across the entire USB device.

According to Mokube (2008) the data recovery methodology used in the research is contrary to manufacturer claims that the Apple® *iPod*® factory restore function wiped or erased data content stored in the device's flash memory. The operating system analysis in the study was again conducted on a Windows® XP computer system. Similar forensic analysis could also be carried out on current and deleted mount point records from Mac OS® X operating systems to establish if a particular iPod® had previously been connected via USB to an Apple® Mac OS® X computer (Kokocinski, 2010).

2.3.3. USB Examination Frameworks

There is no one technical framework developed specifically for the examination of USB devices. Digital forensic practitioners tend to utilize accepted local and international industry best practice approaches. For example, the Scientific Working Group on Digital Evidence – Best Practises for Computer Forensics v2.1 (SWGDE, 2006) and Standard Operating Procedures (SOP) have been developed by organisations to support operational and evidential requirements for general digital forensic examinations.

Sansurooah (2009) highlighted in a recent study of USB flash memory examinations that there are still no frameworks or standardised methodologies for USB examinations compared to other accepted forensic standards that are supported by government agencies and professional standards bodies. According to Vacca and Rudolph (2011), traditional digital forensic frameworks tend to concentrate on critical areas of collection, preservation, analysis and reporting.

Likewise, Beebe (2009) identified that analytical approaches to digital evidence rely on traditional indexing, keyword searching and logical data reviews. Data analysis can be time consuming and could quickly lead to higher data retrieval overheads for practitioners and clients alike when complex systems and multiple devices have to be triaged and reviewed for specific evidential artifacts. To reduce these types of overhead on the practitioner, Beebe further suggested that specific analysis and tool development methodologies must be incorporated into the analysis and reporting areas of new examination frameworks.

Recent industry research on USB forensic examinations has partially addressed the lack of development in USB based analysis frameworks (Lee, 2009). Lee developed standardised examiner guides relevant to Windows® XP, Windows Vista® and Windows® 7 operating systems to allow practitioners to manually record specific Windows® registry locations and USB related artifacts for reporting purposes. These examiner guides could easily be incorporated into an organisation's wider Windows® based examination and analysis framework.

2.4 PROBLEM AREAS AND ISSUES

A number of issues arise from the academic and IT based literature reviews. As discussed in Section 2.2.1, the majority of USB forensic studies reviewed have been based on the popular Windows® XP operating system. Very few studies have explored Windows® 7 based USB research and the development of USB related forensic methodologies due in part to the product's recent release and adoption by technology manufacturers and consumers. No research was located that compared current USB toolsets to assist industry practitioners in selecting software tools that will be beneficial to the examination and reporting of USB storage device artifacts in 2011 and beyond.

Like other ubiquitous storage mediums, there are many different makes and types of USB technologies available today. For example, USB devices such as the *U3* S*mart Drive*® can be both a friend and foe for digital forensic practitioners. According to Al-Zarouni and Al-Hajri (2007) the *U3 Smart Drive*® has built-in technology features over and above normal USB storage devices that allowed both user applications and forensic tools to be seamlessly run on live Windows® XP systems whilst simultaneously storing data on another storage area of the device. These features take advantage of the *AutoRun* feature that is enabled by default in Windows® XP and a virtual read-only International Organization for Standardization (ISO) archive image file. In essence, the *U3* technology tricks the operating system into thinking that both CDROM and USB storage devices are connected to the system at the same time from one device.

U3 technology can also create problems for inexperienced digital forensic practitioners when an examination occurs on a Windows® based forensic workstation. In particular, forensic preservation and data integrity issues may arise if the *AutoRun* feature is left enabled on Windows® XP operating systems or is manually enabled by the practitioner on Windows Vista® and Windows® 7 operating systems for other administrative tasks or system testing prior to an examination of USB connected media items.

There is also a high risk that data on the USB device will be changed if forensic write-blocking devices or software are not used as a standard practise between the forensic workstation and the connected device. The use of industry standard hardware write-blockers often causes some USB devices and external PSDs not to be recognised by Windows® operating systems and forensic software when communication commands are blocked between the workstation and connected device. Specialist software write-blocking tools are recommended in these types of situations to ultimately achieve a forensic review or imaging process without changing data content (Pittman and Shaver, 2010). The write-blocking solution is not infallible and at times can malfunction and stop protecting the connected USB device.

Menz and Bress (2004) argued a similar point in their study of software write protection issues. Write-blocking software is heavily reliant on co-existing within the hosting operating system and as such is just as susceptible to general system instability or catastrophic failure on occasion.

The authors also identified that write-blocking issues can be caused by poor design or incompatible software applications, USB port failures, corrupted drivers, and Windows® updates that may render the software ineffective.

Just as some software applications and write-blockers may cause issues during USB examinations, human or operator error is a challenge that has to be minimised and managed by well-defined processes and standard operating procedures. Errors can be caused by the write-blocking software not being properly installed or tested before an evidence-related USB device is plugged in and examined. Incorrect configuration of registry-related data values may also trigger write-protection functions to be bypassed if software write-blockers are not used as a matter of course (Menz & Bress, 2004). The error is evident if the Windows® Registry key *HKEY_LOCAL_MACHINE\SYSTEM\CurrentControl-Set\Control\StorageDevicePolicies* is not set to the correct *WriteProtect* value of "1" (meaning system write-protection is enabled) and if application validation is not carried out before an examination is conducted by the digital forensics practitioner.

External PSD hard drive examinations also have some challenges when it comes to determining distinct connection artifacts. In a study of PSD forensic analysis Lee (2009) identified that the *ParentIdPrefix* value was no longer recorded in later Windows® operating systems as it had been with standard USB thumb drive examinations in Windows® XP. The study further found that globally unique identifiers (GUIDs) are no longer linked to a device serial number in the *MountedDevices* Registry key on newer Windows® operating systems when compared to similar Windows® XP examinations.

2.4.1. Windows® XP, Windows Vista® and Windows® 7 Issues

The literature review has found a number of issues relating to the Windows® XP operating system that add a layer of ambiguity to such examinations. If a USB device does not contain a valid serial number (*iSerialNumber*) then the Windows® Plug-and-Play Manager will assign a unique identifier number (recognisable by the '&' symbol for the second letter and known as the *ParentIdPrefix* value) to the USB device. According to Carvey and Altheide (2005) the unique identifier value is not as reliable as a device that contains a manufacturer assigned serial number.

Their study demonstrated that if two identical USB devices without manufacturer serial numbers were connected to a Windows® XP system one after the other, the second device adopted some of the first device's recorded GUID string values.

The interpretation and correlation of registry artifacts can cause problems for digital forensic practitioners particularly when dealing with Windows® operating systems that contain a multitude of USB storage device connection history records. The *ParentIdPrefix* value found in the UIID subkey and the USB device serial number can also easily be confused if practitioners do not have an in-depth understanding of how Windows® creates the different types of Registry values (Carvey, 2009).

Equally so, relying on forensic tool reporting output and associated date and time stamps without further validation practises can also cause legal and credibility issues for digital forensic practitioners if the practitioner or evidence output is challenged in court of law. Validation of all processes and output is particularly important when recovered USB artifacts are being produced for evidential purposes. These specific issues will be highlighted in the next subsection.

2.4.2. USB Examination and Reporting Issues

At the present time, a number of traditional forensic software tools support the examination of both USB devices and the recovery of operating system artifacts and connection histories. Both EnCase® Forensic (Guidance Software, 2011) and Forensic Tool Kit® (AccessData, 2011) do support the imaging of USB devices and registry analysis involving Windows® based examinations. However, USB reporting with EnCase® can become very time-consuming and complex when multiple USB devices are involved in an investigation. This is very evident when forensic image copies are manually processed without the aid of third-party processing tools or automated scripting applications. Other freeware forensic tools such as RegRipper© (RegRipper, 2011), USBDeview© (Nirsoft, 2011) and USBDeviceForensics© (Woanware, 2011) support USB device examinations as well but still need to be evaluated in a testing environment to identify standards of functionality, analysis and reporting output.

2.4.3. The Emergence of Anti-forensic Techniques

The final section of the literature review examines the development of anti-forensics practices in relation to USB storage devices. Anti-forensics in the context of the current study can be best described as tools or techniques that are designed to make the analysis of USB evidence decidedly difficult or impossible to conduct.

Bosschert (2006) examined the use of a USB enabled *U3 Smart Drive*® devices on a Windows® XP operating system to identify if certain software tools can conceal USB activity. The manufacturers of U3 based technology claimed that a user would be able to browse the Internet with a Firefox™ web browser and access installed software applications on the U3 device with little trace of user activity. The research conducted by Bosschert determined that U3 related directories, application footprints, SYSTEM hive information, and file name traces were still able to be recovered after a device had been connected to a test computer system.

In contrast, Thomas and Morris (2008) took a completely different research approach by developing an anti-forensics proof of concept software tool called USB M0dY~fire©. Testing of the experimental tool identified that it was possible for a computer user to make changes, falsify device information and even delete current entries related to USB activity in the various keys and subkeys of the Windows® Registry. Whilst the use of these anti-forensic techniques had the power to greatly obscure USB data, the software tool was not infallible as the authors discovered when they overlooked other Windows® Registry entries that were not initially considered during the initial tool design.

The study reinforced the notion that rigorous testing and validation must be used throughout the software development process to identify design-level flaws in the tool before it is released for industry use. This is especially relevant to USB based memory device forensics as there is a potential for the tool to produce incomplete data output or for the practitioner to overlook potential evidence artifacts if all relevant registry and system file locations identified in Table 2.2 are not examined and reported on.

For digital forensic practitioners to be able to detect anti-forensic techniques they must have a detailed knowledge of USB technologies and the location of

relevant artifacts that can be found in any operating system. Furthermore, practitioners need to verify and compare all physical USB storage devices against forensic or device findings to establish that no artifacts or devices have been overlooked or misinterpreted during the overall evidence collection, analysis and reporting phases of an investigation.

The use of visualisation tools to identify coherent relationships between complex evidence items and timestamp variables may be a method in determining whether Windows® Registry files containing USB artifacts have previously been tampered with (Olsson & Boldt, 2009). To explore the merits of anti-forensic detection theories and examination methods, further research needs to be conducted into developing an intuitive timeline function for future USB analysis tools and examination frameworks.

2.5 CONCLUSION

In conclusion, Chapter 2 has presented an overview of the current state of knowledge relating to USB technologies and the importance of digital forensic examinations. The majority of literature reviewed was primarily focused on the examination of Windows® XP operating systems and associated USB artifacts. According to an online globe usage marketing company (Net Market Share), Windows® XP still had more than 52.41% of the operating system market share as opposed to Windows® 7 at 25.89% in early 2011 (Net Applications, 2011a).

While operating systems have evolved in recent years, research has not kept up in the arena of USB system logging or the recovery of all available USB forensic artifacts. Furthermore, the literature review has identified that industry based professionals such as Lee (2009) and Carvey (2011) have successfully exploited some of these voids in a positive manner by providing the global digital forensics community with a number of online and published resources to assist in conducting USB examinations.

From a global perspective the field of digital forensics is still developing as a scientific discipline. This tends to go some way towards explaining why there have been gaps in both USB and Windows® 7 based research in recent years. The literature review also highlighted that there is no standardised frameworks for USB examinations or reporting formats for associated software tools.

To date these software tools are largely based on Windows® Registry extraction techniques that access multiple subkey locations to extract USB related artifacts with few or no visible data validation methods. In terms of tool validation, no specific research was found that concentrated on evaluating a range of tools for use in real-world industry examinations and legal proceedings.

The literature review has identified that there is potential to enhance USB forensic analysis and reporting processes by evaluating a sample set of USB tools and developing a prototype USB reporting tool during the current research project. In order to develop the reporting tool, various software design and tool development methods will be researched to form a research methodology. A USB examination framework will also be developed so that the tool can be evaluated against a sample set of common USB analysis tools used by the New Zealand digital forensics community. The evaluation phase will use empirical data techniques to collect, analyse and validate USB data results in a controlled and forensic environment. These collection methods and an overall research methodology will be defined in Chapter 3.

Chapter 3

Research Methodology

3.0 INTRODUCTION

A researcher's use of an established research methodology is vital for generating reliable research outcomes so as to add to value and knowledge of the subject matter being studied. The chapter aims to discuss and build a valid research methodology in order to answer the hypothesis and associated research questions posed in Section 3.2. It consists of five sections that are structured in a progression, from the review of previously published study methodologies to the development of a research paradigm and design process.

Section 3.1 begins with an overview of different research methodologies and approaches that have previously been used in similar IT, digital forensic, and software development research. Five sample studies were used to establish how the researchers conducted their research and to see what implications there would be if one of the methods were to be used in the current research proposal.

Section 3.2 derives the hypothesis statement and associated research questions from the gaps found in Chapter 2 and the exploration of previous research methodologies found in Section 3.1. A research road map is also developed in Figure 3.1 to visually connect the research questions and methodologies with later sections containing the data analysis, findings and recommendations of the study.

Section 3.3 discusses a design science approach to the overall development of a research framework and identifies five main areas of interest. These areas include problem diagnosis, theory building and hypothesis creation, experiments and tool evaluation, and software design as preferred steps in the current design process. A software development model will also be proposed to facilitate the development of a prototype software tool for USB based memory device forensics.

Section 3.4 outlines the various data collection and analysis techniques (using both quantitative and qualitative methods) that will be used in the analysis and evaluation phases of the research. Presentation of the tool evaluation results and the proposed prototype software tool output will also be discussed.

Finally, Section 3.5 discusses the limitations of the research from two different aspects: the existing sample toolset that will be evaluated as part of the analysis phase and the proposed software tool design that will be discussed in this chapter.

3.1. REVIEW OF PUBLISHED SOFTWARE TOOL STUDIES

The literature review in Chapter 2 identified a number of issues relating to both USB forensic examinations and the general field of digital forensics. The review found that few academic studies have targeted both the Windows® 7 operating system and related USB artifacts. Traditional IT research methodologies are more commonly used by digital forensic researchers and industry professionals to support (or refute) theories when finding solutions to complex real-world software and analysis problems. A number of IT and digital forensic related research studies will be reviewed to identify what approaches and methodologies different researchers have used over a cross-section of tool-based studies. These reviews will also ascertain how each researcher has developed their examination or design frameworks and collection techniques to answer the research findings.

3.1.1. A Windows® System Restore Software Tool Approach

Yun et al. (2008) conducted research into the forensic value of System Restore Point (SRP) analysis on Windows® XP based operating systems. The overall research approach encompassed the theoretical background of Windows® System Restore functions (i.e. reinstallation of critical operating system and application files), the development of a SRP analysis tool and scenario testing to identify and recover related forensic evidence.

An understanding of the subject matter was achieved by conducting Restore Point analysis on Windows® XP system artifacts. These artifacts were identified as the System Volume Information folder, the *change.log* and *rp.log files*. Forensic software tools such as a hex editor were used throughout the research to capture both ASCII and hexadecimal data screenshots of the files for the reader to visually gain an understanding of the theoretical background to each of the artifacts of interest. The overall software development approach taken by the researchers was to design a SRP analysis tool.

A method based on qualitative instrumentation allowed data records to be extracted from the restore points on a "live" computer for further record type classification and analysis. This direct approach would more than likely make system and file changes on a live computer system when the program was executed and any examination actions would need to be meticulously recorded for evidential purposes. Whilst the analysis or interpretation methodology was not clear in the tool design section, a screenshot of the SRP tool in operation does however display detailed restore point data output.

To illustrate the tool implementation, the researchers use a scenario-based examination approach that focused on the Stealth Mailer© program as an example. The program example clearly identified how the tool could analyse various binary and log files relating to SRP activity. The scenario testing successfully identified a sequence of events (more commonly referred in the industry as timelining) relating to the previous deletion of the Stealth Mailer© program by a computer user. The analysis output could then be used for evidential purposes in a real investigation.

The researchers acknowledged that systems analysis was complex in nature and time-consuming when completed under manual conditions. The research also recognised that an automated analysis tool approach was advantageous for digital forensic examinations as long it is rigorously tested throughout the development and productions cycles, and all data output is verified before use.

3.1.2. A Windows® Recycle Bin Software Tool Approach

Gao and Wu (2009) conducted research into the analysis of deleted Recycle Bin data on both Windows® XP and Windows Vista® operating systems. The researchers used existing knowledge gained from previous studies to determine how the Windows® Recycle Bin preserved deleted file artifacts in different operating system versions. Gao and Wu also discussed how an XML data structure model would be used in the research to store recovered data for further analysis purposes.

The researchers moved from theory-based research into a more action-based research methodology by conducting experimental laboratory-based forensic analysis on various Recycle Bin files.

These files included the older Windows® XP *INFO2* file and two newer Windows Vista® files of type *$R* (relating to data) and *$I* (relating to information). These files have replaced the single *INFO2* file format. The files were examined using a popular forensic tool called WINHEX® to identify the fundamental data structures and contents of each file so as to gain a more practical understanding of the overall Windows® recycle process. The analysis techniques and learning methodology used in the study are at the core of everyday digital forensic examination practices and will also be adapted in the current research to establish a baseline of USB connection to a Windows® 7 computer system.

Also incorporated into the research design was the development of a Recycle Bin Forensic Analysis Platform© tool that utilised XML data techniques. The XML based techniques allowed data analysis to be conducted on various system and information files containing deleted file information in a Windows® operating system. The tool was programmed using Visual C++® 2005 with a Windows® Explorer like user interface that incorporated additional progress and output interfaces to visually display current progress updates to the user.

Tool operation screenshots, coding function examples and flowcharts were used throughout the research. These were used to explain and reinforce key aspects of the Windows® file deletion process, along with the design output of the tool platform and the various analysis processes that are needed to produce a data output report. Tool testing was achieved by conducting an experiment that used the contents of a Windows® XP Recycle Bin to showcase the functions of the analysis platform tool.

3.1.3. A Test Station Software Approach

DeAbren (2000) conducted research into the implementation of test software to assist in product development processes. The development approach used a modular software design for the overall software program. A modular design uses smaller parts that interconnect within a larger software package. The researcher identified that a modular design approach provided flexibility in the use of the test software and created general cost benefits in allowing different components to be reused in other testing phases with minimal cost to the total development budget.

The study classifies a module as a separate part of a software program such as a function, process or subroutine. The use of a modular design also allowed for greater independence of lower-level components and solvability if software issues arose without affecting the higher-level components or the software program as a whole. DeAbren (2000) describes the modular design as a "bottom to top" approach that is based on key software design strategies known in the software development industry as top-down (or a step-wise system), and bottom-up (piecing together into a larger system) development methods (Jalote, 2005).

Key components identified in the software architecture included implementation of a user interface, data collection, analysis and logging/test reporting modules. These modules allow control by a user, automated collection and analysis of captured data, and the production of a report at the end of the test. The advantages of this approach are the modules' independence to each other, their portability into other systems, and the ease whereby testing units and debugging software could be embedded into each modular system to enhance the overall software output and module integration.

Other benefits of the design that were discussed in Jalote (2005) included: allowing the creation of different test scenarios with little modification in each module, and the reuse of the modules during the overall development life cycle. The research concluded that the tool design did have some perceived limitations with the additional requirements of more system memory, processing time and some further function calls (i.e. the name of a particular calculation and related arguments). The test software performance was however, not affected by these limitations during testing because the installed memory and processing speeds of the computer systems were able to handle the added load placed on them.

3.1.4. A Mixed-Method Research Approach

Turnbull (2007) used a combination of literature reviews, case studies, data collection and analysis methods to develop a framework design in a study of the computer forensic examinations relating to Wireless Networks (WiFi 802.11) in Australia. The literature review provided an understanding of how wireless networks were exploited for criminal gain whilst at the same time it also provided a gauge for the author to understand the current state of forensic techniques and analysis tools used in such investigations.

The review subsequently found a lack of related tools or up-to-date procedures and established examination frameworks to aid industry practitioners in their examination of wireless devices and associated networks.

Turnbull (2007) adopted a mixed-method approach for his research design. A mixed-method approach uses a combination of both qualitative and quantitative frameworks to drive IT research development. The qualitative aspect of the research was identified by the author as the more influential method of the two because it used case studies, reports and data analysis to answer the hypothesis. There was also no numeric or statistical data recorded during the collection phase of the research to conduct a quantitative comparison for use in further hypothesis testing. Furthermore, Turnbull identified that a mixed-method approach improved data collection techniques and was also well suited for the exploratory nature of wireless and forensics-based research in 2007. The last point is particularly important as very few academic studies had been conducted at the time and wireless forensics was perceived as a relatively new paradigm in the field of digital forensics.

Whilst not a traditional approach, a software development model was also used in conjunction with the mixed-method approach to create a better understanding of the forensic aspects relating to WiFi investigations and to provide a mechanism for the development of new forensic analysis tools (Turnbull, 2007). Whilst not specifically examined by Turnbull, these industry models such as the Classic Life Cycle (CLC) Model (Pressman, 2001) have been widely accepted by the IT industry and provide a proven and structured approach to the software engineering life cycle. The CLC Model approach will be used as part of the current research to assist in developing a USB reporting tool that is based on these standardised industry development practices (Spenser, 2010). Other analysis methodologies utilising both forensic tool evaluations and the recovery of Windows® Registry artifacts (more specifically the *NTUSER.DAT* file and *SOFTWARE* Hive) were also discussed. These Registry artifacts were displayed in screenshot overviews to provide industry practitioners with a range of recovery techniques for use in real-world investigations.

The study proposed both a WiFi forensic processing and investigative model that followed other accepted industry models (such as the CFSAP Model) and best practises at the time (Turnbull, 2007).

Furthermore, the model was designed to allow its framework and examination processes to be easily adapted for any future developments or changes in both Wireless and forensic technologies. Turnbull also recommended that the research would require further development for it to be used in a "live" investigative environment as legal requirements, evidence collection methods and operating procedures are different across both law enforcement and corporate jurisdictions in Australia, and the wider international arena.

3.1.5. An Experimental Research Approach

Liu (2008) conducted a study of Bluetooth network technology by testing performance factors in an indoor-based decentralised (ad hoc) Bluetooth Information Exchange Network (BIEN). The literature review identified that earlier studies had relied on software-based Bluetooth simulators that may have not reflected a true Bluetooth environment nor adequately tested signal effects/loss. These perceived shortcomings were overcome by taking the proactive approach of using a physical Bluetooth network and commercially available Bluetooth products in a more realistic laboratory based testing environment.

Over a one month period, one hundred experiments were carried out with different scenarios and test procedures whilst utilising three Bluetooth networks. These small networks are called *piconets* and contained between three and five Bluetooth devices each to test different traffic and communication patterns. A BIEN software interface was also developed to assist in sending test messages and capturing output. Data output was captured to measure both throughput (i.e. sending and receiving test messages) and latency (i.e. the time taken to send and receive test messages). Quantitative analysis was conducted to compare the different Bluetooth schemas and measurement targets that were used. The quantitative analysis also used statistical charts to draw conclusions from and to answer the hypothesis questions.

The study found that commercially available Bluetooth devices and Bluetooth technology could provide a more realistic environment than traditional software-based simulation by increasing technical and research knowledge of wireless related technology. Liu (2008) recommended that further studies be conducted to develop greater functionality in the user interface of the software

tool. Other recommendations included testing mobile devices such as cell phones and alternate network/security protocols using similar experimental trials in an outdoor setting to increase network performance and security mechanisms in small wireless networks.

3.1.6. Identifying a Preferred Research Methodology

Sections 3.1.1 to 3.1.5 identified a range of research approaches that were previously undertaken by different IT and forensic based researchers. Most scientific or traditional approaches to software and systems-based research have generally taken a quantitative approach using either descriptive or prescriptive methods (Hevner, March Park & Ram, 2004). Understanding the true nature of a technology based environment in order to effectively analyse related systems and data is at the core of descriptive research (March & Smith, 1995). Prescriptive research is about design processes and improvement of product-based knowledge whereas descriptive in comparison is more about traditional research in finding answers to questions (Iivari, 2007).

The differing methods shown in each of the studies does tend to indicate that a broader approach is now being taken by researchers to incorporate other types of research into their research design. These other types can include qualitative or a mixture of both methods, commonly referred to as a mixed-method or hybrid-methods. The mixed-method approach may also utilise non-traditional research methods to further assist in designing a more comprehensive framework for the researcher to work within (Turnbull, 2007). In the general field of IT and especially in digital forensics-based research, a mixed-method approach would certainly be advantageous so a wider range of data collection, analysis and reporting methodologies could be employed by the researcher, when compared to a single-method or more restrictive approach. The current study will therefore adopt a mixed-method approach during the design and data requirement phases of Sections 3.3 and 3.4.

3.2. DEVELOPING THE RESEARCH QUESTIONS AND HYPOTHESIS

In order to define the hypothesis statement and research questions, a review of both IT and forensic related research methods in Section 3.1 and USB related literature in Chapter 2 was needed to define the overall approach of the current research. As stated previously in Section 3.1.6, a mixed-research approach developed by Mingers (2001) coupled with the development of a software analysis and reporting tool will be adapted in the current research project. At a foundation level, the approach combines both quantitative and qualitative research methods. Quantitative research is in the form of laboratory experiments to collect object data from the USB device and Windows® 7 Registry locations whilst qualitative research methods are used to conduct software tool evaluations and content analysis so as to link USB artifacts from an operating system to a particular device.

Chapter 2 identified that the majority of USB research todate had studied older Windows® XP operating system and its associated artifacts. Only a small number of academic and industry based studies such as Alghafli et al. (2010), Lee (2009) and Harvey (2009) had solely concentrated on more modern Windows® operating systems and Registry artifacts such as Windows Vista® and Windows® 7. Section 2.4.2 further identified that forensic tools such as Forensic Tool Kit® (FTK®) and EnCase® Forensic along with Registry-specific toolset functions used by RegRipper© and USBDeview© for example have differing levels of reporting and information output that may or may not be beneficial to the digital forensics practitioner when examining a USB device.

Other problem areas that were also recognised include the use of different types of USB device. For example the *U3 Smart Drive*® adds a layer of complexity to forensic imaging processes and may cause data integrity issues if not handled correctly. Antiforensic software tools/techniques that remove or falsify USB related Windows® Registry artifacts were also highlighted as another problem area. From the literature search, one research problem area with three related components was identified as having the potential for research to improve the analytical outcome of USB forensic examinations. Essentially there is a lack of established USB examination frameworks and the lack of consistent and user-friendly reporting for USB forensic analysis.

This is particularly noticeable in modern Windows® operating systems such as Windows® 7. Furthermore, to date no research was found that compared current toolset functions against a sample of existing forensic software tools to further assist industry practitioners in enhancing the examination and reporting of USB device artifacts. Section 3.2.1 will develop research questions and a hypothesis statement in order to offer a solution to the current problem area.

3.2.1. Main Research Question

The current research aims to explore how the analysis and reporting functions of a forensic software tool can be improved to assist real-world USB investigations. It specifically targets USB storage device technologies (i.e. USB thumbdrives and external hard drive enclosures) that have previously been connected to a Windows® 7 operating system environment. From both a research and forensic standpoint, the USB forensic discipline is still developing when compared to other established data analysis methodologies and research studies that in the majority of cases have heavily focussed on the Windows® XP operating system (Farmer, 2007; Thomas & Morris, 2008; Carvey, 2009).

The research will encompass different aspects from specialised areas that are contained within the IT and digital forensic domains. These areas include USB device technology, forensic methodologies and artefact analysis, software development and data output in order for the hypothesis to be tested and reported in Chapters 4 and 5. Therefore, the main question the research seeks to answer is: **What tool design features improve end-user analysis and reporting of USB forensic artifacts?**

3.2.2. Research Sub-Questions

A number of secondary or sub-questions have been developed in order to assist the researcher in answering the main research question and associated hypothesis. In turn each sub-question aims to provide a connection to the main research question so as to form a better understanding of the different aspects that frame the wider research objective. These sub-questions are as follows:

Sub-Question 1: What is the current state of forensic research related to USB storage devices?

Making a determination on the current state of USB forensic research is essential in being able to identify a particular problem area so as to develop targeted forensic solutions that add value to the wider academic and digital forensics communities. The answer to the sub-question will be found through a review of academic and industry research literature examples that are contained in Chapter 2.

Sub-Question 2: What operating system records are generated by USB activity on a Windows® computer system?

The use of previous study examples found in the literature review from Chapter 2 can form a theoretical understanding of how various Windows® operating system versions record the connection of USB storage devices. Further laboratory experiments will be conducted with a number of different USB storage devices in both the data collection and analysis phases of the research to turn the theory into a more in-depth action-based examination of the Windows® Registry and related USB artifacts.

Sub-Question 3: What specific Windows® 7 Registry evidential related artifacts can assist a forensic practitioner in USB examinations?

In order to build on the theoretical knowledge gained from research sub-question 2, laboratory-based experiments will be conducted using various USB storage devices (both thumb drive and portable storage drives) and a test computer system installed with a fresh installation of the Windows® 7 operating system. Using the latest operating system version will allow the researcher to identify what registry locations and potential evidential artifacts have changed from previous research involving the more established Windows® XP operating system. There will also be particular focus on whether recorded Windows GUID values can be used by a practitioner to distinguish the type of USB hard drive enclosure that has previously been attached to a Windows® 7 system. Little research, apart from Lee (2009) has been conducted in the particular area. The answers to the sub-question will also allow for a more comprehensive USB analysis checklist to be developed for use in industry based laboratories.

Sub-Question 4: What forensic or commercial tools examples are currently available to the examiner for collecting and reporting on USB artifacts?

The literature review in Chapter 2 identified a number of existing prototypes and commercially available software tools that have been used for USB related research and forensic examinations in the past. Four such tools will be chosen for evaluation purposes after consulting a representative sample of industry practitioners to determine what existing tools are frequently used for USB forensic examinations in New Zealand. Further evaluation testing will be conducted to determine what strengths and weaknesses each tool has with particular focus on the collection and reporting functionalities. The aim of the tool evaluation is also to identify potential gaps in a sample of current tool offerings that could be resolved by the creation of a specific USB analysis and reporting prototype tool for Windows® 7 related USB examinations.

Sub-Question 5: What key tool features could be incorporated into the proposed tool design to benefit future USB forensic examinations?

A key outcome of the research is to determine if a collection of reporting mechanism can improve the output and workflow of modern USB examinations. The proposed prototype design is likely to incorporate both device and data analysis to allow a practitioner to conduct a comparative analysis of collected forensic artifacts and *device descriptor* information. The proposed prototype tool will be evaluated against some existing tools to identify if these new design features are an improvement over existing tool designs.

Sub-Question 6: What protection mechanisms need to be incorporated into the proposed tool design for reliable data output?

The answer to this research sub-question will come through the use of an established software development model and common validation or mathematical hashing techniques to prevent data from being altered or contaminated during the tool processing and reporting phases. Further testing and data analysis during the evaluation phase will also be used to identify if data output is understandable, accurate, and complete.

Sub-question 7: What improvements does the proposed software reporting tool need to have on existing USB forensic tools and recovery techniques?

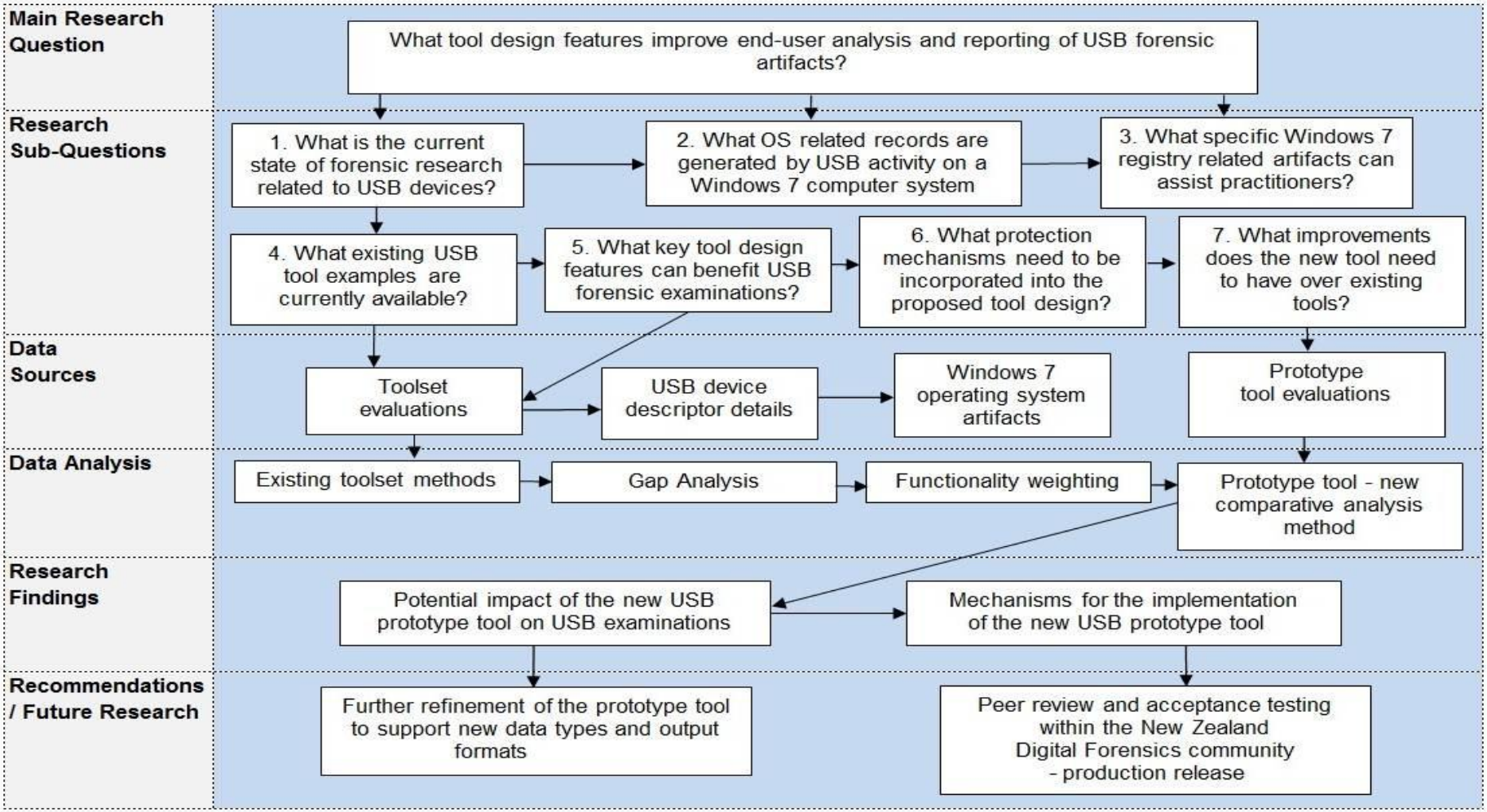
In order to answer this sub-question a testing framework and templates will be developed. Data from the tool evaluations will be collated and analysed to determine what potential benefits the new prototype tool has over a sample set of frequently used USB forensic tools within the local digital forensics community. A benefit analysis will explore areas such as usability, tool processing time, consistency of artifacts collection, reporting formats and cost to establish if the new prototype tool can improve the reporting output of existing USB tool examples.

3.2.3. The Proposed Hypothesis

The researcher proposes a hypothesis that related to the development of a Windows-based forensic prototype tool for the analysis of USB evidential information. The tool aims to simplify the collection of USB-related artifacts in order to increase the quality of industry-based forensic evidence reporting. The collection process will be based on the extraction of both operating system and device artifacts from a standardised and widely used evidence file format (i.e. the EnCase® E01 evidence file). The reporting process will be based on the analysis of targeted Registry data from a single computer system to produce detailed and consistent output reporting for use by industry practitioners.

The prototype software tool will solely focus on USB related Windows® 7 operating system artifacts, as anecdotally it is a commonly used operating system that is encountered by New Zealand digital forensics laboratories at the present time (May 2011). Therefore, the hypothesis statement for the current research is formulated as follows: **USB digital forensics examinations are improved by enhancing the reporting capability of software tools**. Figure 3.1 offers a road map for the current research. It provides a summary of the research questions and successive methodologies that will be used during the course of the research to investigate the hypothesis. Section 3.3 will discuss a research design that will be developed to allow the hypothesis statement and related research questions to be tested through a combination of USB-related laboratory experiments, data collection and analysis methods.

Figure 3.1. Research Road Map



3.3. RESEARCH DESIGN

USB storage device analysis like other systems based analysis is frequently a complex and time-consuming task for digital forensic practitioners to conduct (Yun et al., 2008). Currently there is no one complete extraction or reporting tool solution that is available to assist practitioners in the analysis of USB artifacts. In order to address the particular problem area, IT based design science methods and a combination of both traditional and non-traditional research principles will be used by the researcher to develop a research design framework and a set of methodologies for the project. These components will then be used to develop and evaluate a reporting solution that addresses current challenges faced by examiners when conducting the forensic examination of USB devices.

3.3.1. A Design Science Approach

A design science approach forms the basis of technology-based research by allowing an IT researcher or industry professional to apply problem-solving techniques to improve information systems, processes and related activities (Venable, 2006). Two essential design science processes that have been identified by March and Smith (1995) are creating and evaluating software to assist in the problem solving process. The aim of the current research design is to examine USB based memory storage device activity on modern Windows® operating systems with the assistance of a new and experimental software reporting tool. The prototype tool will be compared against other forensic software applications and recovery methodologies in order to find a solution that will improve the discovery and reporting of USB device artifacts.

In order to implement a design science approach and accomplish the aims of the research, a framework is developed and proposed to assist in identifying and visualising the different areas and processes that form part of this research. Figure 3.2 illustrates how the formulation of a hypothesis from gaps found in previous USB research studies is central to the core areas of problem diagnosis, software design, field experiments and tool evaluations.

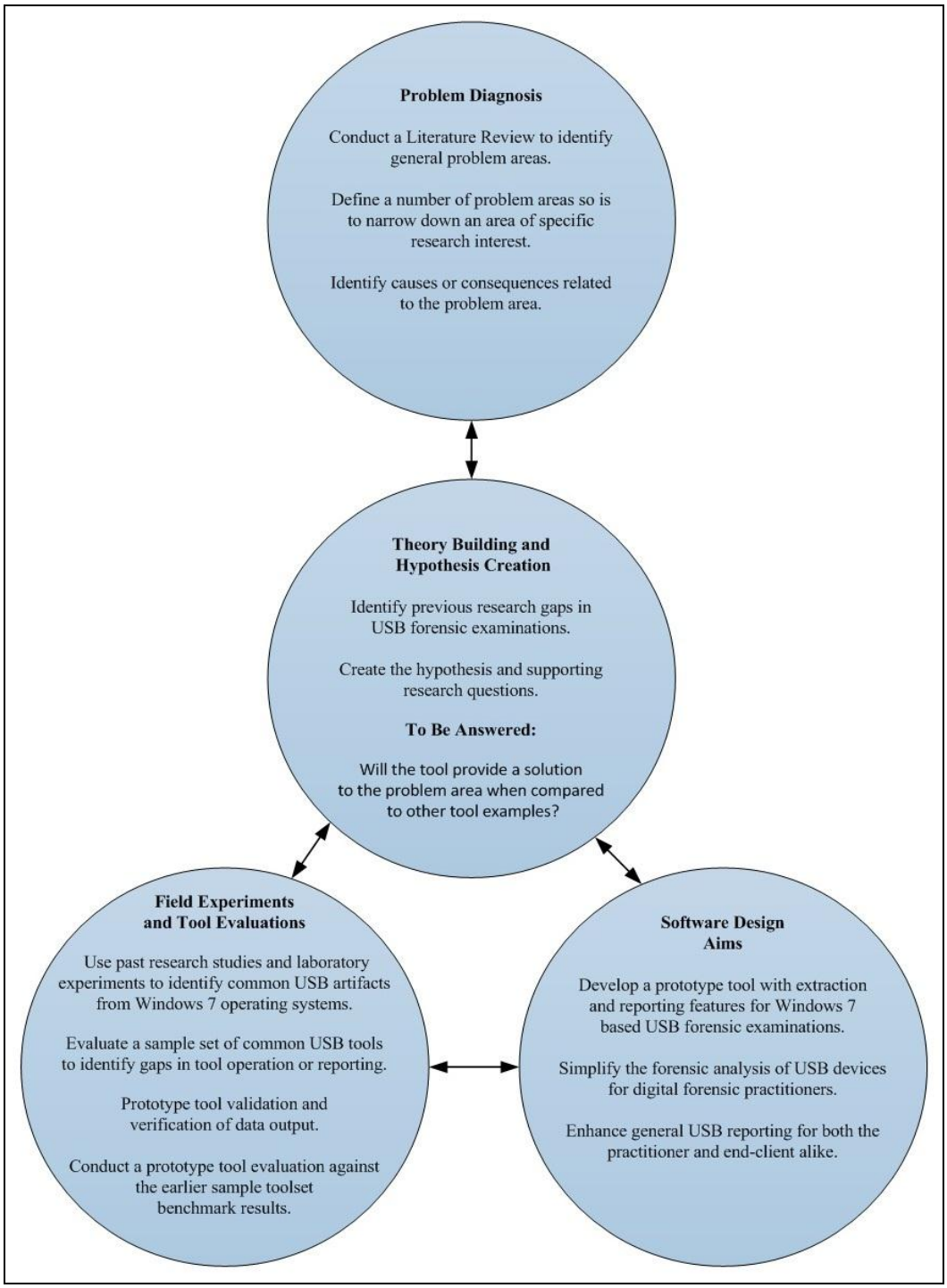


Figure 3.2. A Design Science USB Framework for the Current Research Project. Adapted from the Role of Theory and Theorising in Design Science Research, by J.R.Venable, 2006. In the *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology, DESRIST 2006*, Claremont, CA, p.17. doi=10.1.1.110.2475

The proposed technology-orientated research framework is therefore based on established design science principals.

The following subsection will outline a series of strategies that will be used to provide assistance in finding answers to the problem domain.

3.3.2. The Research Methodology

The representative forensic and tool development studies discussed in Chapter 2 and Section 3.1 indicated that previous USB and IT based research had used different types of research approaches and methodologies. The current research is about testing the hypothesis through the mixture of both positivist and interpretivist methods that were previously utilised by earlier IS research studies such as Mingers (2001). Positivist research uses instruments for collecting data, laboratory experiments, and data analysis whilst interpretivist research uses qualitative content analysis and descriptive grouping or timelining of data to test theories and solve problem areas.

Section 3.1.6 proposed that mixed-method approach be taken in the current research using a combination of traditional quantitative and qualitative methodologies and a non-traditional approach using software development to test and answer both the hypothesis and associated research questions. Quantitative research methods will be employed by the researcher through the use of common forensic tools such as FTK® Imager and EnCase Forensic® software to collect Windows® 7 based operating system and USB storage device data for further analysis. Qualitative methods will make use of experimental observations and collected data objects to interpret how a Windows® 7 operating system records USB related artifacts. Data output from the new software tool will also be interpreted along with data from a representation of other available USB forensic tools to evaluate how each performs so as to establish if the new tool has improved USB examination processes.

In operational terms, the methodology approach that will be used for the research will concentrate on five main phases:

Phase 1: USB device and forensic related theory.

Phase 2: Sample toolset evaluations, dataset collection and tool analysis.

Phase 3: Gap analysis for performance and functionality.

Phase 4: USB tool development – prototype software created.

Phase 5: Prototype tool validations, field testing and benchmark comparison analysis.

The central themes of the methodology are driven by toolset evaluations, laboratory experiments (involving data collection and data analysis) and software development. Data collection will be accomplished through the use of existing forensic software techniques in controlled laboratory-based experiments. The controlled experiments and forensic collection techniques will enable detailed analysis of USB device and Windows® operating system artifacts so that answers can be found for the first three research sub-questions. The creation of a USB reporting tool and its evaluation against a sample of exiting forensic software tools will further assist in finding answers to the remaining four research sub-questions and main research question. This development process will also enable the researcher to prove or refute the research hypothesis.

3.3.3. The Software Design and Tool Testing Methodology

A primary outcome of the research is the development of a prototype software tool that will ultimately be designed to meet the needs of local digital forensic practitioners conducting Windows® based USB examinations. In section 3.1.6 it was mentioned that software design is not traditionally associated with conventional scientific or established IT based research methodologies. Software design does however offer a technical problem-solving mechanism and may contribute to the body of knowledge in the particular field of research. Therefore, outcomes that are comparable to the more traditional approaches of theoretical and action-based research can also be achieved through the combination of both theory and practice in a complementary manner (Andriessen, 2007).

The software design of the current research will make use of established software engineering or development models that are frequently used in IT based research and within the IT industry to develop software tools (Craig & Jaskiel, 2002; NIST, 2009; Gao, 2010). Pressman (2001) and Spenser (2010) identified a number of widely recognised Software Development Life Cycle (SDLC) models, each with their own advantages and disadvantages. The most popular models and associated variations are:

- The Classic Life Cycle (CLC Model). Also known as the Linear Sequential or Waterfall Models;
- The Prototyping Model;
- The Rapid Application Development (RAD) Model;
- The Component Assembly or Component-Based Development (CBD) Models.

The use of proven SDLC models is advantageous from both a research and industry perceptive as they are widely accepted in a professional sense. The frameworks underpinning each model have well-defined structures and require process and software verification along with consistency of design and output (Pressman, 2001). From a forensic standpoint of view, the verification of software processes and data output are considered essential components to the overall forensic process (Beckett & Slay, 2007). Data validation and output accuracy are especially relevant when evidence is presented in either a civil or criminal judicial hearing, or when forensic processes are subject to scrutiny by opposing counsel and other digital forensic experts.

The specific model chosen for the current research is the CLC Model that incorporates some elements of the RAD Model. The prototyping model was not selected for the current research study because client evaluation and feedback prior to any production release of the proposed tool has been designated as part of future research, and therefore remained outside the current research scope. The main advantage of the CLC model is that it has been widely used and has been revised over time (Ruparelia, 2010). The predefined and sequential steps of the model also allowed each development stage to be validated and quality tested before the next is started. The main disadvantage of selecting a model to use at the research proposal stage was the uncertainty of implementing an SDLC model without having previously used formal software development models in other research projects or real-world software development assignments.

The CLC Model is driven by an ordered set of well-defined development phases that can be simplified in the context of the current research study to the following six generalised areas of design:

- A set of key software requirements;
- Analysis structure and extraction staging;
- Code generation to interpret the required Windows Registry Hives and associated string values;
- Process and output debug testing;
- Tool testing and validation against similar tool samples;
- Live laboratory experiments using various USB thumbdrives and external storage devices.

The RAD Model is a modified version of the CLC Model that allows the software development cycle to be implemented in a quicker timeframe without compromising the core design or final output (NIST, 2009). The main reasons for using elements from this particular model are that the current framework is component-based and also there is a limited development timeframe so the software tool testing and evaluation phases can be commenced within the time parameters of the research project. Some coding and software components such as the identification and extraction of the Windows® Registry strings from evidence datasets may also be publically available and in a standardised code format so new extraction functions or processes will not have to be recoded from the ground up.

Like in any technology-based product development, unexpected software and hardware issues may be encountered at any time during the development cycle. Lehman and Sharma (2011) identified that CLC Model variants are reliant on core delivery requirements which can be affected by unplanned delays or other production variables. Therefore, the proposed research methodology and software design conditions may have to be modified to assist in the completion of the prototype software tool. Figure 3.3 shows the combining of the CLC and RAD development models to assist in the overall software design process.

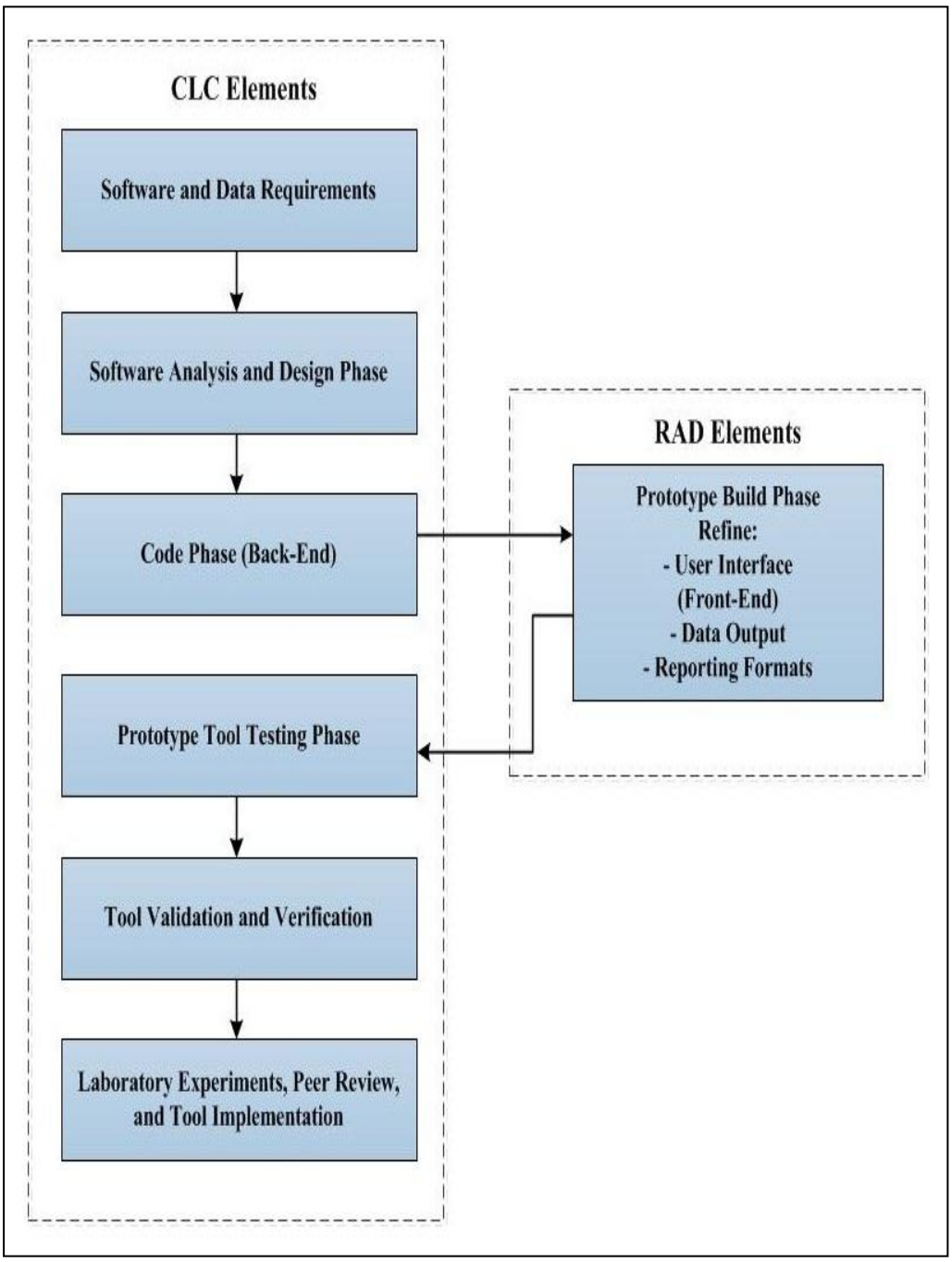


Figure 3.3. Proposed Software Design Using CLC and RAD Model Elements

The interface for the proposed prototype tool will be complied using aspects of the C++ programming language. The C++ language is a general-purpose programming code that has been popular with software developers for designing both computer system and portable application programs. The selection of the particular programming language does have some function limitations and complexities but it is a useful language given the research time-frame.

The tool interface will be user-friendly in design and simple to use for the end-user. For the most part all extraction and processing functions are automated and the practitioner is only required to make minimal selections. These selections are to choose the required evidence set, start the extraction and analysis processes, and save a log or an analysis report.

The logging and reporting options will be presented in a HTML file format for standardised importation into an analysis report and/or in technical notes for legal disclosure purposes. Figure 3.4 displays the general tool flowchart functions that will be developed using this type of the design approach. Each of the functions may be subject to modification throughout the design process and could even change as the underlying processes are coded, and or tested.

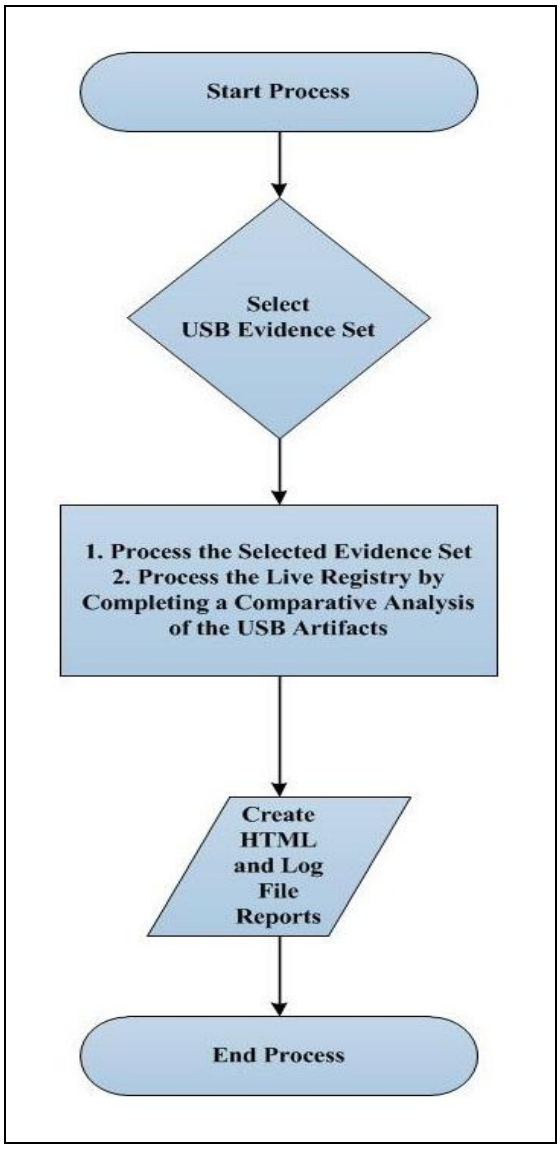


Figure 3.4. Proposed USB Flow Chart for the Functioning of the Prototype Tool

The preferred tool evaluation methodology will use a tool testing template that was developed by the researcher as shown in Table 3.1. The proposed template is based on both NIST (2005) and SWGDE (2009) tool testing/validation recommendations that have been widely accepted and utilised by the international digital forensics community. The evaluation criteria for the tool evaluations has also been adapted from established NIST testing requirements to suit the current testing environment. There are individual eight *conditional requirements* developed as part of the evaluation criteria, and each has the same weighting as the other (Refer to each of the completed templates in Appendix C for full details).

Table 3.1

Proposed Toolset Evaluation Template for the Evaluation Sheets

Test Name and Device Information	
Test Details	A unique identification number is designated for each test sequence. USB device details and the overall test actions are also provided in more detail.
Tester	The name of the person conducting the testing action.
Test Date(s)	The date(s) the specific testing sequence was conducted.
Conditional Requirements	The predefined evaluation criteria known as conditional requirements (CR1 to CR8) which are applicable to the current toolset evaluations.
Source and Destination Hard Drive Information	Source (in this case the suspect hard drive) and destination details (in this case the designated evidence hard drive). Details can include but are not limited to the make, model, serial number, total sector count, and the interface for each storage device.
Forensic Image Hash Value	The actual hash type and value produced during the forensic imaging process.
Post Analysis Forensic Image Hash Value	The actual hash type and value produced after the toolset evaluations are completed or at any time during the testing and analysis processes when there is a need for the evidence file to be verified.
Sample Toolset Details	Sample tool name details, software version, developer details, general usage or licence restrictions and any additional software requirements required.
Logging and Exported Data	Logging and bookmark export information noted for each test sequence. Full printouts will be provided in Appendix C.
Tool Results	The conditional requirements are met or not met for each test sequence.
Test Outcomes and Comments	Individual tool results of note where conditional requirements have not been met, or have only been partially met. Further details relating to software errors or anomalies in tool operation can also be recorded by the tester.

The proposed research requirements and final template layout will be subject to modification and/or redesign as the research develops. Any such changes will be decided once the sample toolset capabilities have been evaluated in a laboratory environment.

3.4. DATA REQUIREMENTS

Section 3.4 will discuss the different methodologies and data requirements to be used in the current research. The section also identifies how the different types of data to be collected will fit into the various experimental and evaluation areas of the research. These areas are classified in the next three subsections as data collection, data analysis, and data presentation.

3.4.1. Data Collection

From a forensic perspective, evidential data can be collected from a computer system via four common sources: the operating system, categorised by Bell and Boddington (2010) as "live" or "dead" analysis capture, installed applications, attached storage devices such as a conventional hard drive or USB storage drives, and remote network storage locations. Data collection in the current research will be achieved in a laboratory environment by connecting four USB thumb drives and two external USB storage hard drives (of varying makes and storage capacities) to identical versions of the Windows® 7 Home Premium operating system. The proposed collection method will provide a baseline for further data analysis and also simulates normal everyday usage for all of the USB product types on a Windows® based computer system.

The new operating system installations will each be achieved by using VMware® virtualisation software to provide pre-configured, consistent and repeatable testing environments. The tool evaluation method will then use a series of eight tests for each individual tool. A total of eight tests (forming 48 datasets) are proposed by using this methodology. The virtualised hard drive environments and physical USB devices will then be forensically imaged (i.e. a bit-for-bit physical copy) after each tool evaluation series using FTK Imager® forensic software in accordance with industry best practice to capture the associated data in an evidential manner and to provide enough data content for analysis in Section 3.4.2.

Both individual tool logging output and a research journal will be utilised throughout the collection phase to maintain a record of all actions that the researcher has taken. Like other established forensic procedures, the journal will contain sufficient details such as laboratory equipment used, forensic software versions, operating system and USB device information, and collection methodologies so as to allow the research to be validated or replicated by another third party in the future.

The expectation of the data collection phase is that all USB storage devices will be forensically imaged as per standard forensic examination and evidential conditions so the collected data can be analysed without changing the original device content. Any device, software tool or write-blocker issues will be examined, reported and then resolved using other acceptable forensic tools and methods so the next analysis and presentation phases can be completed.

3.4.2. Data Processing and Analysis Methods

Data processing will be accomplished by using the various logging outputs collected from the toolset evaluations identified in Section 3.4.1. Data will then be entered into the individual evaluation sheets as shown by the template in Table 3.1. The evaluation sheets and relevant entries in the research journal will in turn be summarised in a spreadsheet evaluation matrix that contains a summary of the individual tool details, conditional requirements and "pass" or "fail" graded results. The evaluation matrix will then be used as a basis for the data analysis phase of the research.

Data analysis is an integral part of any scientific process. From the perspective of the digital forensics discipline, data analysis is about associating artifacts with processes and sources that created them (Andrew, 2007). In the current research, vendor and device information will be extracted from the forensic image copies of each USB test device to determine what source or vendor created identifiers are available on each device. An in-depth critical analysis will also be conducted on the Windows® Registry and associated system files of each forensic hard drive copy. The analysis will substantiate the registry sources discussed in the literature review of Chapter 2 and establish what forensic artifacts and footprints are left by USB usage on a Windows® 7 based operating system.

Table 3.2 classifies the different data types and locations where forensic extraction and analysis techniques will be concentrated on during the toolset evaluations and analysis phases.

Table 3.2

Data Types and Reference Locations of Forensic Value

Data Type	**Actual Data or String Location**
Registry	*\HKLM\SYSTEM\CurrentControlSet\Enum\USB*
Registry	*\HKLM\SYSTEM\CurrentControlSet\Enum\ USBSTOR*
Registry	*\HKLM\SYSTEM\MountedDevices*
Registry	*\HKLM\SYSTEM\CurrentControlSet\Control\Deviceclasses\ {53f56307-b6bf-11d0-94f2-00a0c91efb8b}*
Registry	*HKLM\Software\Microsoft\Windows Portable Devices\Devices*
System	*\Windows\inf\setupapi.dev.log*
User	*\Users*UserProfileName\NTUSER.DAT\Software\Microsoft\ Windows\CurrentVersion\Explorer\MountPoints2*

To further assist in determining the strengths and weaknesses of individual tools used in the toolset evaluations, a gap analysis is proposed as part of the current research methodology. According to the Institute for Security, Technology, and Society at Dartmouth College, a gap analysis provides a way "to determine where gaps in product availability may exist" (ISTS, 2004, p.5). The proposed gap analysis will use the evaluation matrix results to make a determination of the overall performance of the evaluated tools against the test requirements. This form of gap analysis also has the advantage of easily identifying weaknesses in exiting tools so new analysis or reporting features can be incorporated into the prototype USB tool currently being developed as part of the research project.

3.4.3. Data Presentation

The final phase of the forensic process model identified by NIST (2006) involves reporting the results. Reporting in forensic examination terms can be best described as the process of arranging and presenting data from the analysis phase for further review or legal scrutiny. The aim of any research report or evidence presentation is to provide the intended audience with data output that is accurate, concise and structured in a manner that makes it easily read and understandable.

The tool evaluation results will be presented in a series of tables and charts to visualise the research findings for easy interpretation.

Examples of the proposed presentation methods could include conditional requirements and associated "pass" or "fail" field results for each tool being displayed in a matrix with associated colour coding. Bar graph charts could also be used to plot and visually compare average processing times across all of the evaluated tools. Both methods would also provide an equally detailed and graphical indication of overall tool performance during the toolset evaluations.

The data presentation for the prototype tool will be in the form of the widely supported .html output. The .html data output allows for electronic disclosure and ease of reader access through the use of common web-browsing software such as Internet Explorer® or Firefox®. The html data format is commonly used by other forensic software programs and the source code has an added advantage that it can easily be changed with an html/text editor to suit different reporting or webpage layouts. As previously noted in Section 3.3.3, the data output format may also have to be further refined or changed once the initial prototype testing and data collection phases are completed.

3.5. LIMITATIONS AND EXPECTED OUTCOMES

The research will have a number of limitations relating to the evaluation of third party tools and the proposed USB prototype tool. Furthermore, the overall scope of the research is limited to Windows® 7 operating system artifacts as Chapter 2 clearly demonstrated that previous research has largely centred on Windows® XP based forensics examinations. There will also be opportunities beyond the scope of the research for further refinement of the new prototype tool to support non-Windows examinations such as those from Apple® Macintosh® operating systems.

3.5.1. Limitations of the Sample USB Tool Evaluations

The USB tool evaluations will be limited to a sample selection of existing IT and forensic tools that are currently used for USB examinations in New Zealand. There are many registry and USB device tools that are freely available for both IT and digital forensic practitioners to utilise in network administration and forensic related duties. It is also envisaged that the tools will various limitations in functionality, forensic image support and reporting formats. These issues will

have to be factored into the evaluation testing by the researcher to identify if the selected tools are suitable for further testing.

A recent informal phone survey conducted by the researcher determined that only four USB tools are commonly used by a sample selection of law enforcement and corporate practitioners in New Zealand. These tools are identified as EnCase Forensic®, FTK® RegistryViewer, USBDeviceForensics© and USBDeview©. Two of the tools (USBDeviceForensics© and USBDeview©) have not been previously used by the researcher and will be blind-tested along with the others to simulate normal software usage cycles in a digital forensics laboratory.

The USB device selection will consist of four separate USB thumb drives and two external USB hard drive enclosures of varying storage capacities. Each was picked at random and covers a wide range of vendors. The evaluation testing will only occur over a three week period to allow for further refinement of the new prototype tool. Limiting the range of USB storage devices and the length of the evaluation period will still allow for quality data to be collected for analysis purposes without incurring too many repetitive experiments that produce similar content and outputs (Liu, 2008).

3.5.2. Limitations of the New Software Prototype Tool

The new prototype software tool is limited in general terms to specific extraction and time-lining of Windows® 7 based USB artifacts from a forensic image copy. Only the Registry hives and associated string values identified in Table 3.2 will be analysed and reported in the following analysis and findings sections. Initial testing of the prototype tool may also reveal a number of processes or functions that need further coding and refinement. Any specific limitations that are identified during the testing will also be discussed in Chapter 6.

3.5.3. Expected Research Outcomes

The forecasted outcome of the current research is the development of a prototype USB software tool to enhance the forensic analysis and reporting capability of law enforcement and corporate practitioners. It is anticipated that the overall hypothesis and supporting research questions will be answered by evaluating an existing sample set of Windows-based USB analysis and forensic tools against the new prototype.

Like any forensic practise or process, the integrity and quality of the research largely relies on a detailed understanding of the subject matter using a combination of both theoretical and practical applications. It is envisaged that the practical tool development and evaluation phases of the research will assist digital forensic practitioners in choosing a wider range of recovery and reporting techniques for USB based memory device forensics.

3.6. CONCLUSION

The purpose of Chapter 3 was to connect previous USB knowledge and theory presented in the literature review of Chapter 2 with an overall design approach and software development methods that will be employed in the current research. The current research design has been achieved by using a mixed-method approach that best suits the core digital forensics and software development aims of the research. The overall methodology uses a design science framework approach with traditional quantitative and qualitative data gathering methods. A non-traditional software development model is also utilised so toolset evaluations and laboratory based experiments can be conducted for further analysis in Chapter 4.

The creation of an appropriate methodology will move the research from a design phase into a more experimental and analysis driven phase that culminates in the creation of a new prototype software tool for further testing and evaluation. The expected outcomes of the research have been discussed and will be compared against the toolset findings presented in Chapter 4. The limitations of the research have also been identified and will be routinely assessed as the tool development and evaluation phases near their final completion.

Chapter 4 will present the data analysis and toolset evaluation results to ultimately identify what impact the new prototype software tool could have on future Windows-based USB forensic examinations within the New Zealand digital forensics industry.

Chapter 4

Research Findings

4.0 INTRODUCTION

The research methodology established in Chapter 3 incorporated elements of forensic functionality testing and software development. This enables field evaluations to be conducted using a sample set of common USB analysis tools. Establishing the capabilities of a particular forensic tool or toolset under field conditions is critical if the resulting data is presented as digital evidence in criminal or civil proceedings (Ryan & Shpantzer, 2005). This chapter provides a benchmark for USB tool capability and performance whilst also filling the USB tool evaluation gap identified in the literature review of Chapter 2.

Chapter 4 comprises of four sections and reports on data collected during the tool evaluations to provide findings for the field research. Section 4.1 addresses a number of modifications to the data collection and processing requirements of the research methodology formulated in Chapter 3. Section 4.2 presents and explains the data collected for the toolset benchmarking that was needed to facilitate the data requirements identified in Chapter 3. The collected data will be analysed in Section 4.3 using a comparison or gap analysis matrix to identify individual tool performance against the conditional requirements derived in Section 3.3.3. Furthermore, the gap analysis matrix also provides a platform to identify potential analysis and reporting issues in the evaluated toolset so improvements can be incorporated into the tool development process.

Finally, Section 4.4 delivers a summary of the major research findings the sample toolset evaluations to provide a baseline for further analysis and data presentation using the developed USB tool in Chapter 5. The findings in Chapters 4 and 5 will also provide the foundation for discussion of the research results in Chapter 6 and for answering the current hypothesis formulated in Section 3.2.3.

4.1. MODIFICATIONS TO THE DATA REQUIREMENTS

At the outset of the testing phase, a number of technical and tool capability issues were encountered and modifications had to be made to the original data requirements specified in Section 3.4. The technical issues relate to the testing environment and data collection methodologies. These issues were unforeseen at the time the initial testing methodology was being developed. The issues and modifications are discussed in the following the three sub-sections before the main field findings are reported in Section 4.2.

4.1.1. Testing Environment

Section 3.4.1 determined that a virtualised operating system environment would be implemented in the field testing phase. A Windows® 7 based virtual machine template was created as a consistent test operating system platform using VMware® virtualisation software. A further seven templates labelled Test 1 to Test 7 were then created from the original template for each sample USB device to be tested with no technical issues being encountered.

Preliminary testing of the Test 1 virtual machine platform identified that the USBDeview© tool was predominantly designed to be used in a "live" operating system environment (via local or remote connection). Further research discovered that the tool had a command-line capability to analyse extracted *SYSTEM* registry files. The command-line capability was added to the tool testing criteria for the USBDeview© evaluation in Section 4.2. The initial testing also identified that the *USBDeviceForensics*© sample tool failed to run correctly in the virtual machine test environment without the relevant .Net Windows® Updates being applied. The resulting error made it impossible to process the registry and system data and caused the researcher to quit the user interface. Figure 4.1 depicts the error window after the *System Hive, ntuser.dat* and *Setupapi.dev.log* files had been loaded into the program and processing was commenced.

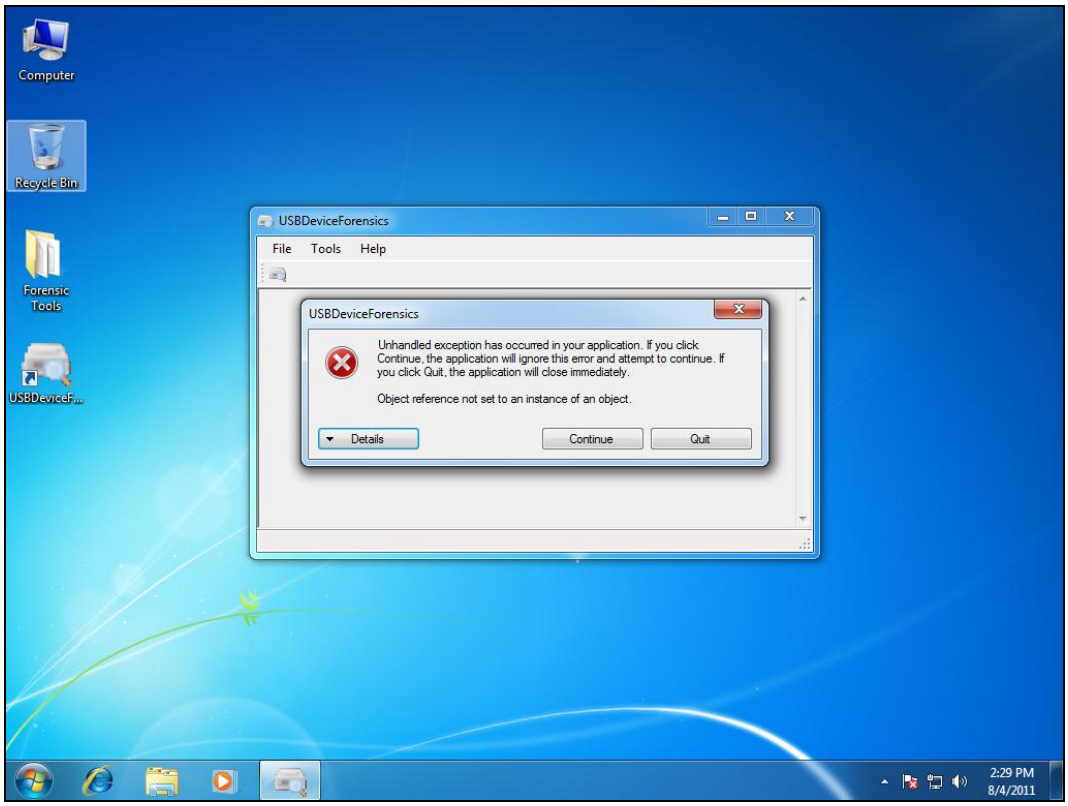


Figure 4.1. USBDeviceForensics© Error in a Virtualised Test Environment Encountered by the Researcher

As a result of the issues encountered, the baseline testing environment was changed to a more stabilised physical host operating system platform so that all tools in the sample toolset could be run without further technical issues being encountered. At the completion of the new Windows® 7 operating system installation and software configurations, the test hard drive was forensically imaged to enable the current field test environment to be recreated for future audit or research validation purposes.

To keep the consistency of a baseline test image, disk imaging software named Macrium Reflect™ (Standard, v5) by Paramount Software UK Limited (Paramount Software, 2011) was used to restore a common image for each individual test scenario so the same Windows® 7 operating system and software application configurations were applied throughout the tool evaluation phase.

4.1.2. Data Collection

The final modification to the proposed data requirements methodology involved the decision by the researcher not to forensically copy each of the USB devices being used in the sample tool evaluations.

Instead *device descriptor* information from the firmware of each device was captured for later analysis and validation purposes. The discovery of *device descriptor* research by Microsoft® (2010) in Section 2.1.2 of the literature review coupled with initial field imaging of a sample USB device supported the decision-making process. The *device descriptor* information contains significant vendor, product and serial number artifacts that are embedded within the non-volatile memory or firmware of the USB device. Field sampling determined that forensic software does not capture the embedded hardware information and furthermore, no specific manufacturer artifacts were located in the data storage area of a sample USB device that could identify a particular USB device for forensic examination purposes.

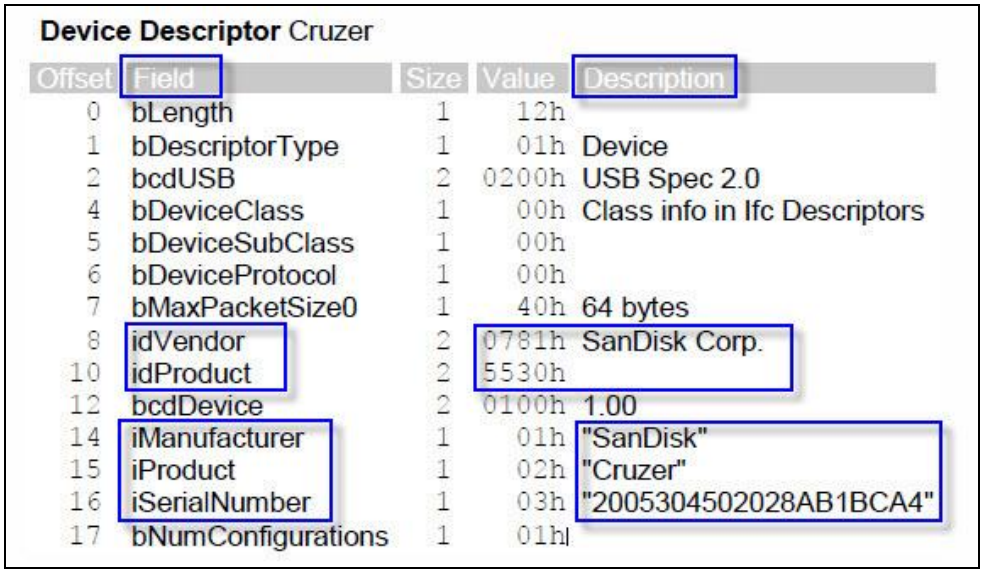
Commercial software applications such as USBlyzer© (USBlyzer, 2011) can display the *device descriptor* information of a USB device being examined. USBlyzer© is a Windows® based USB software protocol analyser that allows USB device activities to be captured and analysed on a running computer system. USBlyzer© was used in the research to capture *device descriptor* information of each USB device under forensic conditions (i.e. through the use of hardware or software write-blocking technologies). Figure 4.2 displays a screenshot of the USBlyzer© software that captured USB *device descriptor* properties from a SanDisk Cruzer USB device used in the current sample toolset evaluations. Full *device descriptor* information for all of the test USB devices is reported in Appendix C.

Device Descriptor Cruzer

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	0781h	SanDisk Corp.
10	idProduct	2	5530h	
12	bcdDevice	2	0100h	1.00
14	iManufacturer	1	01h	"SanDisk"
15	iProduct	1	02h	"Cruzer"
16	iSerialNumber	1	03h	"2005304502028AB1BCA4"
17	bNumConfigurations	1	01h	

Figure 4.2. Captured USB Device Descriptor Information by USBlyzer©

Figure 4.3 highlights the various *device descriptor* field and description values that will provide a specific baseline reference for the USB storage devices used in the tool evaluations. These values will be validated during the benchmark testing phase to assist in determining if each tool accurately records the embedded *device descriptor* information for a particular USB device.



Device Descriptor Cruzer

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	0781h	SanDisk Corp.
10	idProduct	2	5530h	
12	bcdDevice	2	0100h	1.00
14	iManufacturer	1	01h	"SanDisk"
15	iProduct	1	02h	"Cruzer"
16	iSerialNumber	1	03h	"2005304502028AB1BCA4"
17	bNumConfigurations	1	01h	

Figure 4.3. Specific Device Descriptor Information of Interest captured by USBlyzer©

4.1.3. Data Processing and Analysis

The data processing and analysis phases were completed as outlined in Section 3.4.2. Only specific USB data artifacts from the various registry hive and system files highlighted in Table 3.2 were analysed for reporting in Chapters 4 and 5.

4.2. BENCHMARK TESTING OF THE SAMPLE USB TOOLSET

In order to develop a prototype USB reporting tool and ultimately test the main hypothesis statement, the capabilities of a sample set of existing USB analysis and reporting tools needed to be evaluated. Benchmarking the sample toolset against specific evaluation criteria identified in the current research as Conditional Requirements (CR) has the potential to determine a level of overall performance and examination output for commonly used USB tools that have not been reported before. Likewise, the toolset evaluations also have the ability to pinpoint potential areas of improvement in analysis and reporting functions that could be incorporated into a new USB prototype tool under development.

Sample toolset benchmarking was conducted in the field using three phases: Phase One outlines the base computer, operating system configurations and general application settings for each of the seven testing scenarios; Phase Two summarises the evaluation of four commonly used USB analysis tools whilst Phase Three compares the sample toolset against the testing scenarios, and specific reporting functionality of each tool.

4.2.1. Test Phase One: Configurations and General Setup

The baseline computer configuration for the eight testing scenarios followed a standardised methodology that is based on established NIST hardware and software reporting requirements shown in Figure 4.4.

Test Hardware Details

Custom PC workstation containing:
Intel Pentium Dual CPU E2200 2.20 GHz
Installed RAM: 4.00 Gb DDR2 DIMM
Gigabyte S-Series GA-G31M-S2L
REV: 1.1 Motherboard
6 x USB 2.0 ports
1 x PCIe card with 2 x USB 3.0 Ports
Provision for 4 x SATA hard drives

Operating System Details

Windows® Operating System Edition:
Windows® 7 Home Premium (Version 6.1)
32-Bit software - standard installation
Computer Host Name: Test-PC
Workgroup: Workgroup
User Account Name: Test

Figure 4.4. Test-PC Computer Hardware Components and System Details

The hard drive configurations used in the baseline computer system are shown in Figure 4.5.

The test operating system hard drive consisted of a small Windows® 7 operating system on the primary partition (i.e. *Disk 2 [2AEEB5C4] ST3500418AS – Partition 2 – C:*) to allow the baseline image to be imaged and restored within a minimal time period for each of the tests scenarios.

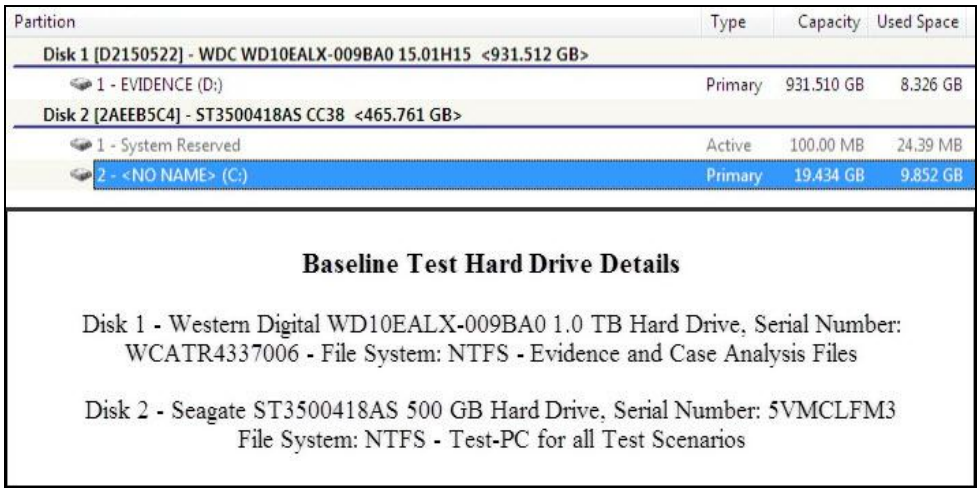


Figure 4.5. Test Hard Drive Configuration and Hardware Details (Adapted from Macrium Reflect, 2011 output)

The hard disk labelled "*Disk 1 [D2150522] –WDC WD10EALX-009BA0 - Partition 1 - Evidence*" displayed in Figure 4.5 was utilised as an evidence and case repository during the collection, processing and analysis phases. All evidence, case files and sample data reports were further backed up to an additional hard drive storage device for safekeeping. This backup procedure is in keeping with standard industry best practise and enables two copies of the evidence files and test data to be kept in case one set is damaged or destroyed during the research project.

A number of supporting tools such as USBlyzer©, Macrium Reflect™ and FTK® Imager Lite were also used to establish the baseline testing environment, imaging and restore processes during each of the evaluation tests. Standardising environment configurations allowed each of the test scenarios to be accurately replicated and also assisted in providing basic information for completion of a research journal in Appendix B. All of the tools and licence requirements utilised during the current research are reported in Table 4.1.

Table 4.1

Testing and Analysis Software Requirements for Toolset Evaluations

Software Name	Version	Purpose and Licence Requirements
USBlyzer©	v 2.0	USB device software protocol analyser - Fully functional 33-day trial for evaluations purposes – no charge for trial download otherwise licence use only. Allowed the *device descriptor* for each USB test device to be captured.
Macrium Reflect®	v 5	Backup and hard drive imaging software - Standard commercial licence used. Provided a standardised testing environment for all evaluation tests.
Microsoft - USBDeview©	v 1.91	USB device utility – Freeware. Sample Tool 1.
Woanware - USBDeviceForensics©	v 1.0.7	USB analysis and reporting software - Freeware. Sample Tool 2.
Guidance Software - EnCase Forensic©	v 6.18	Computer forensics imaging and analysis software – Commercial licence used. Sample Tool 3.
USB Device History EnScript®	v 0.5	Automated EnCase® supported USB device analysis script from Lance Mueller's forensic website: http://www.forensickb.com/ - Free. Used in conjunction with Sample Tool 3.
AccessData Forensic Toolkit (FTK®) and RegistryViewer®	FTK® v 3.3 RegistryViewer® v 1.6.3	Forensic Toolkit® computer forensics analysis software – Commercial licence used. RegistryViewer® software - Limited Demo version or commercial licence. Jointly used for Sample Tool 4.
AccessData FTK Imager Lite®	v 2.9.0	Data review and imaging software – Freeware. Used for forensic imaging of the hard drive data from all of the evaluation tests.

Table 4.2 presents a summary of the USB devices used in conjunction with the sample USB tools during each of the toolset and prototype tool evaluations. The devices were categorised by their product purpose for each test sequence. Tests 1 to 4 used the USB storage devices that had been categorised as USB thumb drives, whilst Tests 5 to 8 used the USB devices that had been categorised as PSD devices. All of the selected USB storage devices support the more commonly used USB 2.0 specification. The Seagate 500 GB FreeAgent GoFlex PSD device is a relatively new hybrid external USB hard drive combining support for the USB 2.0 specification and also the more recently updated 3.0 specification.

Table 4.2

USB Devices Used During the Toolset Evaluations

USB Device Name	**Device Category**
SanDisk 4 GB Cruzer USB 2.0 Flash Drive	USB Thumb Drive
Kingston 4 GB DataTraveler 101 USB 2.0 Flash Drive	USB Thumb Drive
Apacer AH3255 4 GB USB 2.0 Flash Drive	USB Thumb Drive
Dick Smith 2 GB USB 2.0 Micro Drive	USB Thumb Drive
Transcend StoreJet 500 GB USB 2.0 Portable Storage Device	PSD Device
Seagate 500 GB FreeAgent GoFlex USB 2.0/3.0 Portable Storage Device	PSD Device

4.2.2. Test Phase Two: Sample Toolset Evaluations

Phase Two evaluated the four tools as part of a sample toolset to collect sufficient data so as to form the basis for further analysis in Section 4.3. Eight Windows® 7-based test environments (labelled Test 1 to Test 8) were created with each test concentrating on one particular USB device at a time. Each of the four USB tools was individually evaluated on a *Pass* or *Fail* basis to determine the tool's capability and performance against the conditional requirements developed in Section 3.3.3 and provided in Appendix C. The resulting datasets (48 in total) were reviewed and key portions of the outcomes are examined in the following subsections 4.2.2.1 to 4.2.2.8.

4.2.2.1. Test 1: SanDisk 4 GB Cruzer USB 2.0 Flash Drive

Test 1 evaluated the connection of a SanDisk 4GB Cruzer flash drive to a Windows® 7 operating system and computer hardware that supported a USB 2.0 specification environment in order to determine if each of the four evaluated tools was able to meet the conditional requirements (CR) criteria. The test found that USBDeview© tool was unable to meet CR 4 as the tool only supports extraction and analysis of USB artifacts from the *SYSTEM* hive. The USBDeviceForensics© tool did not support reporting of the *Windows Portable Devices* sub-Key information from the *SOFTWARE* hive and therefore was also unable to meet CR4.

4.2.2.2. Test 2: Kingston 4 GB DataTraveler 101 USB 2.0 Flash Drive

Test 2 connected a Kingston 4GB DataTraveler 101 USB flash drive to a Windows® 7 operating system environment using USB 2.0 supported hardware in order to determine if each of the four evaluated tools were able to meet the conditional requirements criteria. The test found that the *USBDeview*© and *USBDeviceForensics*© tools were unable to meet CR 4 as earlier indicated in Test 1. Likewise, *EnCase® Forensic* was unable to meet CR 6 during Test 2 as the tool operation suffered from application hang-ups and a number of failures when moving between each mounted registry subkey in the application interface.

4.2.2.3. Test 3: Apacer AH325 4 GB USB 2.0 Flash Drive

Test 3 connected a Apacer AH325 4GB USB flash drive to a Windows® 7 operating system environment using USB 2.0 supported hardware in order to determine if each of the four evaluated tools was able to meet the conditional requirements criteria. Apart from the *USBDeview*© and *USBDeviceForensics*© tools not being able to meet CR 4, all of the other tools passed each of the requirements.

4.2.2.4. Test 4: Dick Smith 2 GB USB 2.0 Micro Drive

Test 4 connected a Dick Smith 2 GB USB Micro drive to a Windows® 7 operating system environment using USB 2.0 supported hardware. The test was conducted in order to determine if each of the four evaluated tools was able to meet the conditional requirements criteria. With the exception of the *USBDeview*© and *USBDeviceForensics*© tools not being able to meet CR 4, all of the other tools passed the requirements.

4.2.2.5. Test 5: Transcend StoreJet 500 GB USB 2.0 PSD Device

Test 5 connected the first of two USB portable external hard drives, a Transcend StoreJet 500 GB PSD to a Windows® 7 operating system environment supporting USB 2.0 hardware in order to determine if each of the four evaluated tools was able to meet the conditional requirements criteria. With the exception of the *USBDeview*© tool not being able to meet CR4, all of the other tools passed the requirements.

4.2.2.6. Test 6: Seagate FreeAgent GoFlex 500 GB USB 2.0 PSD Device – Scenario 1 (S1)

Test 6 connected the second of two USB portable hard drives, a Seagate 500 GB FreeAgent USB 2.0/3.0 PSD device to a Windows® 7 operating system environment. The evaluation testing was conducted to determine if each of the four evaluated tools was able to meet the developed conditional requirements. The Seagate FreeAgent PSD device was used in the first of three scenarios to create a baseline of normal USB 2.0 and 3.0 device connection conditions with Scenario 1 providing the first time the device was connected to the test Windows® 7 operating system under USB 2.0 specification conditions. The other two scenarios were completed in Test 7 and Test 8 to examine changes in Windows® Registry data when a different USB port was used on another date with the same device, and to test the same device in a USB 3.0 environment. With the exception of the *USBDeview*© tool not being able to meet CR 4, all of the other tools passed the requirements.

4.2.2.7. Test 7: Seagate 500 GB FreeAgent USB 2.0 PSD Device – Scenario 2 (S2)

Test 7 connected the same Seagate 500 GB FreeAgent USB 2.0/3.0 PSD Device as used in Test 6 to a Windows® 7 operating system environment to test the conditional requirements criteria. In the second of three scenarios the device was reconnected under USB 2.0 conditions on a different date whilst utilising a different USB port location. The use of a different port was to establish what if any changes were made to the original registry sub-key entries that had previously been recorded in Test 6.

Identifying changes in related data values provided the researcher with a better understanding of how the Windows® Registry recorded and updated USB information after the initial connection time. With the exception of the *USBDeview*© tool not being able to meet CR4, all of the other tools passed the requirements.

4.2.2.8. Test 8: Seagate 500 GB FreeAgent USB 3.0 PSD Device – Scenario 3 (S3)

Test 8 provided the final test in the benchmark toolset evaluation and connected the same Seagate 500 GB FreeAgent PSD that was used in Test 6 and Test 7 to a Windows® 7 operating system environment in order to determine if each of the four evaluated tools was able to meet the conditional requirements criteria. In the last of the three scenarios the PSD device was connected under USB 3.0 conditions only (i.e. via a PCIe USB 3.0 card port on the test computer system) to establish if the capture of *device descriptor* and Registry artifacts had changed in any way from the older USB 2.0 specification.

The *USBDeview*© tool failed to meet any of the conditional requirements for Test 8 as the Seagate FreeAgent PSD was not detected by the software utility resulting in no evaluation data being captured. Test 8 continued without further technical issues and the other tool evaluations passed all of the conditional requirements.

4.2.3. Test Phase Three: Sample Toolset Evaluation Overview

Forensic software analysis and reporting tools are not created equal. Each tool will perform differently depending on variables such as the original coding methodology, the quality of coding, configuration and environment settings, and understanding of operation or actions by the user. The sample toolset evaluation was conducted under consistent hardware, software and testing environments as outlined in Section 4.1.1.

Only the USB related analysis and reporting functionality of each tool was evaluated so as to take into consideration the different limitations in application functionally and reporting formats of the selected toolset. The results of the sample toolset evaluation are summarised in Table 4.3 and provide an overview of each tool when compared against the specific conditional requirements developed in Section 3.3.3.

Table: 4.3

Sample Toolset Evaluation Results

Test 1			Test 2			Test 3			Test 4		
Tool	**Results**	**Outcome**	**Tool**	**Results**	**Outcome**	**Tool**	**Results**	**Outcome**	**Tool**	**Results**	**Outcome**
Tool 1	87.5%	Fail – CR 4 Partial USB Artifacts Only	**Tool 1**	87.5%	Fail – CR 4 Partial USB Artifacts Only	**Tool 1**	87.5%	Fail – CR 4 Partial USB Artifacts Only	**Tool 1**	87.5%	Fail – CR 4 Partial USB Artifacts Only
Tool 2	93.75%	Fail – CR4	**Tool 2**	93.75%	Fail – CR4	**Tool 2**	93.75%	Fail – CR4	**Tool 2**	93.75%	Fail – CR4
Tool 3	100%	Pass	**Tool 3**	93.75%	Fail – CR6 Application Issues Not Reported	**Tool 3**	100%	Pass	**Tool 3**	100%	Pass
Tool 4	100%	Pass	**Tool 4**	100%	Pass	**Tool 4**	100%	Pass	**Tool 4**	100%	Pass

Test 5			Test 6			Test 7			Test 8		
Tool	**Results**	**Outcome**	**Tool**	**Results**	**Outcome**	**Tool**	**Results**	**Outcome**	**Tool**	**Results**	**Outcome**
Tool 1	87.5%	Fail – CR 4 Partial USB Artifacts Only	**Tool 1**	87.5%	Fail – CR 4 Partial USB Artifacts Only	**Tool 1**	87.5%	Fail – CR 4 Partial USB Artifacts Only	**Tool 1**	0%	Failed to the Detect the USB 3.0 Device
Tool 2	100%	Pass	**Tool 2**	100%	Pass	**Tool 2**	100%	Pass	**Tool 2**	100%	Pass
Tool 3	100%	Pass	**Tool 3**	100%	Pass	**Tool 3**	100%	Pass	**Tool 3**	100%	Pass
Tool 4	100%	Pass	**Tool 4**	100 %	Pass	**Tool 4**	100%	Pass	**Tool 4**	100%	Pass

Designated Toolset Names

Tool 1 – USBDeview **Tool 2** – USBDeviceForensics **Tool 3** – EnCase Forensics **Tool 4** – FTK/FTK RegistryViewer

4.3. ANALYSIS OF THE OVERALL TOOLSET PERFORMANCE

Section 4.2 provided a summary of the toolset evaluation results. Section 4.3 conducts an analysis of the overall performance of each tool using the gap analysis methodology discussed in Section 4.3.1 and combining the results with the field analysis and reporting experiences of the researcher in Section 4.3.2. The field analysis and researcher's experiences take the form of tool observations, journal entries and log information extracted during each evaluation scenario. Section 4.3.3 will use the results from both analysis methodologies to identify gaps in the sample toolset so improvements can be implemented in the design and functionality of the USBForensicReporter© prototype tool.

4.3.1. Sample Toolset Evaluation Gap Analysis: Phase One

A gap analysis methodology was selected in Section 3.4.2 to provide assistance in making a determination of the software tool's capabilities against predetermined analysis and reporting requirements (in this case defined as conditional requirements) of a nature similar to the NIST testing requirements discussed in Chapter 3. TechTarget (2006) provides an appropriate gap analysis definition that is relevant to the current research as being: "an assessment tool to help identify differences between information systems or applications" (para. 1). Gap analysis has previously been used as an analysis methodology in both forensic and IT related software design, quality assurance and tool evaluations (ISTS, 2004; Hoffman & Deal, 2008 and Amaral & Faria, 2010).

The gap analysis involved two phases. Phase One was a combination of both the individual and overall tool analysis from Section 4.2.2 whilst utilising the conditional requirements criteria across the eight benchmark toolset evaluations to allow for the construction of two gap analysis matrices in Tables 4.4 and 4.5. Phase Two concentrated on the analysis and reporting functions of each evaluated tool to identify associated weaknesses in tool operation or functionality. The gap analysis methodology for the current research is shown in Figure 4.6.

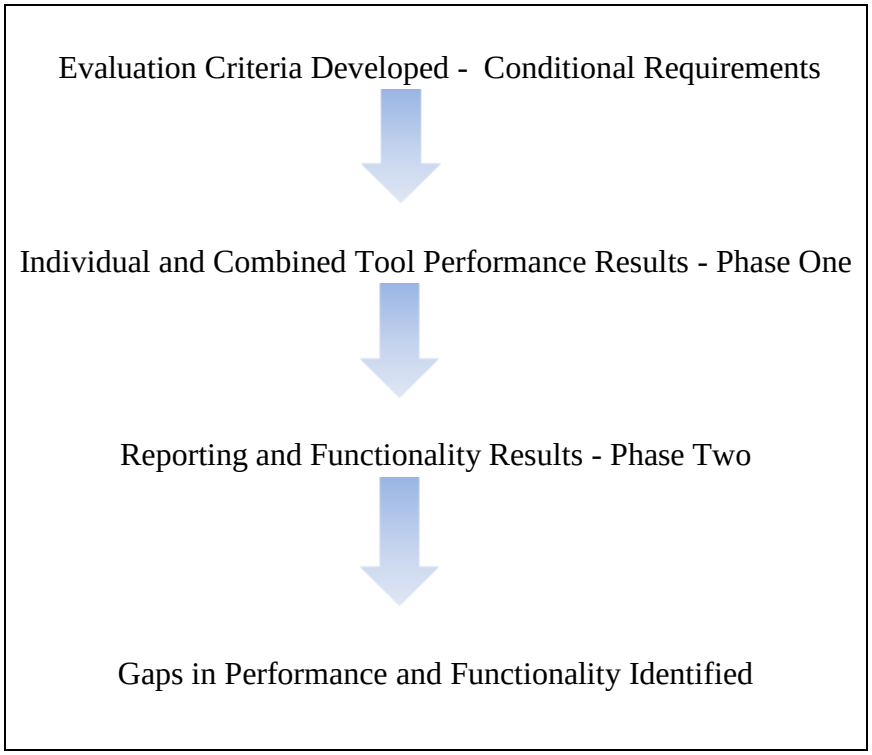


Figure 4.6. Phases One and Two Gap Analysis Methodology Applied for Identifying Gaps in the Sample USB Toolset

The Phase One Gap Analysis results shown in Table 4.3 demonstrate that the *FTK*® tool combination received the highest number of pass marks with an average of 100% when evaluated against the eight conditional requirements being utilised during the toolset evaluations. EnCase® Forensic also achieved an average of 99.22% across the eight test scenarios. Unexplained application hang-ups were experienced by EnCase® Forensic in Test 2 resulting in the manual analysis time being extended considerably when compared to the tool's overall high performance results (an average of 100%) in the seven remaining test scenarios.

The USBDeview© tool consistently failed on CR4 across seven of the eight test scenarios as the analysis and reporting functions of the tool only supported data output from the *SYSTEM* Registry Hive. Also, the tool did not support USB 3.0 device analysis, resulting in Test 8 not being completed. Despite the tool's lack of capability to analyse USB 3.0 devices and other Windows® Registry and system files such as the *NTUSER.DAT* and *setupai.dev.log* file, *USBDeview*© still yielded enough data output to assist a digital forensic practitioner in providing basic reporting on a specific USB device.

The data output from the tool primarily takes the form of *USBSTOR* and *USB* Registry key entries. The entries comprise *device descriptor* and last connection date and time stamp values associated to past USB device connections. Overall the USBDeview© tool evaluation achieved an average performance result of 87.5% across the seven scenarios when USB 2.0 supported devices were being tested. The field evaluations further determined that the most common extraction and analysis methods employed by the sample toolset was the use of forensic image files or offline Registry file examinations.

Both the EnCase® and *FTK*® tools support their respective forensic image file formats to allow a digital forensic practitioner to manually or automatically process the relevant system and Registry files for USB artifact reporting. The USBDeviceForensics© tool is primarily an offline Registry examination tool. Additional extraction measures needed to be employed to allow the Registry and system files from a forensic image or mounted hard drive to be analysed and reported. These measures were easily achieved in the field by using the freely available *FTK® Imager Lite* forensic software to load the forensic image file or to mount a device for the relevant files to be extracted. The *Export Hash List*, *Create Disk Image* and *Export Files* features of *FTK® Imager Lite* were also utilised during the sample toolset evaluations to preserve and collect the data in a forensically sound manner.

The computed hash values (utilising industry standard MD5 and SHA1 hash algorithms) for each extracted Registry and system file were then used after each test scenario to verify that the data integrity of the files had not been altered in any way by the processing functions of each tool. The overall Phase One Gap Analysis matrix results are shown in two matrices (Tables 4.4 and 4.5).

4.3.2. Sample Toolset Evaluation Gap Analysis: Phase Two

The Phase Two Gap Analysis analysed the research journal entries and individual datasets from the toolset evaluations conducted in Section 4.2. Identifying the presence of potential analysis and reporting performance gaps in the sample toolset was advantageous to the current tool design process therefore allowing improvements in the analysis and reporting functions to be made to the new USB reporting prototype tool currently under development.

To assist with the gap analysis, six category themes were devised from the perspective of a digital forensics practitioner using forensic tools in the field, and the researcher's recorded experiences during the recent tool evaluations. The gap analysis categories are defined as *Overall Ease of Tool Analysis* and *Reporting Functions*; *Total Time Taken to Analyse and Report USB Artifacts*; *All Common USB Artifacts Reported*; *Availability of Reporting Formats*; *General Output Quality*, and *Tool Output Needs Additional Action or Formatting*. Each designated sample tool was then ranked against the six categories to determine the presence of any significant gaps.

The grading criteria for the Phase Two Gap Analysis utilised a series of numbered results ranging from *0* to *10*. Score results were assessed against an ideal baseline of measurement for further comparison and reporting purposes. Analysis and reporting functionality categories classified with a grade of *0* represented an overall **poor** performance in functionality, a grading of *5* represented an **acceptable** performance in functionality and a grading of *10* represented the **highest** or **ideal** performance in functionality to the tool user.

The results in Table 4.6 show that all of the sample tools were easy to use with a mean score of *8* out of *10* being achieved for the *Overall Ease of Tool Analysis and Reporting Functions* category.

Table: 4.4

Toolset Evaluation: Phase One Gap Analysis Matrix for USB 2.0 Supported Devices

Conditional Requirements (Testing Criteria)	**Sample USB Toolset for Evaluation**			
	USBDeview	**USBDevice - Forensics**	**EnCase Forensic**	**FTK/FTK Registry Viewer**
CR1 - The tool supports processing of digital source evidence (i.e. evidence file or individual Live and Offline Registry Hive data)	✓	✓	✓	✓
CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard	✓	✓	✓	✓
CR3 - The tool supports the examination and reporting of USB 2.0 devices	✓	✓	✓	✓
CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool	~	~	✓	✓
CR5 - All original digital source evidence is unchanged by any subsequent tool process or user actions	✓	✓	✓	✓
CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user	**N/A**	**N/A**	✘	**N/A**
CR7 - If the tool logs processing information, the information is accurately recorded in a log file and or screen output for the user	✓	✓	✓	✓
CR8 - The tool allows extraction of analysis and log information into a format that is viewable by the user	✓	✓	✓	✓

✓ The tool fully satisfies the conditional requirements testing criteria

~ The tool partially satisfies the conditional requirements criteria

✘ The tool does not satisfy the conditional requirements testing criteria

N/A The tool does not support the conditional requirements or it was not encountered

Table: 4.5

Toolset Evaluation: Phase One Gap Analysis Matrix for a USB 3.0 Supported PSD Device

Conditional Requirements (Testing Criteria)	**Sample USB Toolset for Evaluation**			
	USBDeview	**USBDevice - Forensics**	**EnCase Forensic**	**FTK/FTK Registry Viewer**
CR1 - The tool supports processing of digital source evidence (i.e. evidence file format or individual Live and Offline Registry Hive data)	✘	✓	✓	✓
CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard	✘	✓	✓	✓
CR3 - The tool supports the examination and reporting of USB 2.0 devices	✘	✓	✓	✓
CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool	✘	✓	✓	✓
CR5 - All original digital source evidence is unchanged by any subsequent tool process or user actions	✘	✓	✓	✓
CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user	✘	**N/A**	**N/A**	**N/A**
CR7 - If the tool logs processing information, the information is accurately recorded in a log file and or screen output for the user	✘	✓	✓	✓
CR8 - The tool allows extraction of analysis and log information into a format that is viewable by the user	✘	✓	✓	✓

✓ The tool fully satisfies the conditional requirements testing criteria

✘ The tool does not satisfy the conditional requirements testing criteria

~ The tool partially satisfies the conditional requirements criteria

N/A The tool does not support the conditional requirements or not encountered

Table: 4.6

Toolset Analysis and Reporting Functionality: Phase Two Gap Analysis Results

Gap Analysis Categories	**USBDeview**	**USBDevice-Forensics**	**EnCase**	**FTK Tools**	**Ideal Baseline Rating**
Overall Ease of Tool Analysis and Reporting Functions	8	8	8	8	**10**
Total Time Taken to Analyse and Report USB Artifacts	9	7	6	7	**10**
All Common USB Artifact Groups are Reported	5	8	9	9	**10**
Availability of Reporting Formats	8	8	8	5	**9**
General Output Quality	8	8	8	8	**10**
Tool Output Needs Additional Action or Formatting	9	5	5	8	**10**

Individual tool interface and supporting functions were straightforward to operate and apart from the application hang-ups experienced with the EnCase® Forensic tool in Test 2, no further technical issues were encountered during the evaluations. The results from the *Total Time Taken to Analyse and Report USB Artifacts* category revealed that the USBDeview© tool was the quickest tool to provide analysis and reporting results to the tool user with a score of *9* out of *10*. The USBDeview© tool would be the most appropriate tool for making an immediately determination of basic USB *device descriptor* information from a USB 2.0 device. The mean score for this category was 7.25 out of 10.

The results from category entitled *All Common USB Artifact Groups are Reported* determined that the EnCase® Forensic and FTK® tools scored the highest across the sample toolset with a consistent score of *9* out of *10* being awarded to both tools. The highest scoring tools were able to consistently report on USB artifacts (where recorded) in all of the Registry and system locations identified in Table 3.3. The USBDeviceForensics© tool scored 8 out of 10 as the tool does not support the reporting of *USB* artifacts in the *SOFTWARE* subkeys.

The USBDeview© tool scored an acceptable grade of *5* as the tool only supports analysis and reporting of *SYSTEM* hive related USB artifacts. The mean score for this category was *7.75* out of *10*.

The USBDeview©, USBDeviceForensics© and EnCase® tools achieved the highest results (*8* out of *9*) in the category entitled *Availability of Reporting Formats*, whilst FTK® scored the lowest (*5* out of *9*) as the FTK® RegistryViewer portion of the toolset only permits exporting using a predefined HTML report output. The FTK® RegistryViewer style of report is not easily modified for appending to existing analysis reports and is best suited for standalone or supplementary reports. The reporting feature severely restricted an otherwise outstanding USB forensic tool. The mean score for this category was *7.25* out of *10*. The evaluation results for the *General Output Quality* category were consistently reported in the high range across all of the tools within the sample toolset. No significant reporting issues relating to the raw data output reports were identified. The mean score for this category was *8* out of *10*.

Finally, the *Tool Output Reporting Needs Additional Action or Formatting* category results identified that two tools, USBDeview© and FTK® achieved the highest scores (9 and 8 out of *10* respectively) as little editing of resulting data output was required. Both the USBDeviceForensics© and EnCase® Forensic tools received a lower but acceptable score of *5* out of *10* due to the extra formatting action required to make the exported data output more presentable. The USBDeviceForensics© tool generated reports in both CSV and text file formats; however, the CSV format needed additional exporting and data manipulation for final reporting data purposes.

The EnCase® Forensic tool also provided a number of different methods for generating bookmark reports including text files, RTF and HTML formats. Bookmarks are specific references relating to potential items of interest for investigators to review such as files, folder structures and information, sections of highlighted text and practitioner notes or comments about a particular item. Both text and RTF based bookmark reports exported by the EnCase® Forensic tool required additional customisation and formatting to deliver presentable data for further reporting purposes. The mean score for this category was *6.75* out of *10*.

4.3.3. Gaps Identified by the Sample Toolset Evaluations

The Phase Two Gap Analysis identified three significant gaps in the analysis and reporting functionality from the toolset evaluations. The gaps are important enough to provide the basis for further design and enhancement of analysis and reporting features related to the USB prototype tool currently under development. The first significant gap area was identified in the *All Common USB Artifact Groups are Reported* category. The gap specifically related to the USBDeview© tool as the user interface is currently restricted to analysing and reporting of data from the *SYSTEM* Hive on USB 2.0 supported devices. In the case of a legal proceeding, a digital forensic practitioner must be able to reach conclusions and form expert opinion using tools that provide the most in-depth and accurate analysis of previously connected USB devices on a Windows® 7 operating system. Therefore, for completeness of reporting, data from the *SOFTWARE* hive file, *NTUSER.DAT* and *Setupapi.dev.log* files would need to be included in any forensic tool that is used in extraction and analysis of USB forensic artifacts.

The next significant gap area was identified in the category entitled *Availability of Reporting Formats*. The gap was particularly noticeable in the *FTK® RegistryViewer* tool as the tool's export interface only provided one reporting format. The HTML reporting format consisted of a predefined and branded template that was not easily adapted for additional reporting or editing purposes. The sample toolset evaluations identified that for a tool to report USB artifacts in a professional and easily understood format at least two reporting options must be made available for the user to select. The most suitable reporting options would therefore include text or log file output and HTML reporting so the source HTML coded template could easily be modified using an HTML editor or simple text editor such as Windows® Notepad.

The third significant gap area was identified in the category *Tool Output Needs Additional Action or Formatting*. Reporting gaps were predominantly noticeable for the USBDeviceForensics© and EnCase® Forensic tool results. The CSV export format for the USBDeviceForensics© tool required additional importation into a Microsoft® Excel® spread sheet and further data processing to obtain a fully functional output report.

In the case of EnCase® Forensic, the text and RTF bookmark reports required additional editing and formatting to obtain a presentable report. The use of an output report requiring little or no additional user action would be advantageous to the design of the HTML reporting formats for use in the USBForensicReporter© tool. Therefore, the new USB prototype tool will provide both a text file format for logging and analysis output along with a modifiable HTML template that is readable and meaningful in layout. Both reporting formats will allow subsequent data output to be printed, exported and disclosed in a straightforward and simplified manner.

A Microsoft® Office 2010 Excel® Radar chart type was chosen to display the Phase Two Gap Analysis results in Figure 4.7. The Radar chart style provides a visual concentration of strengths and witnesses for each evaluated tool using the six individual gap analysis categories verses the grading criteria developed in Section 4.3.2. The three black indicator arrows represent the most significant gaps from the outer ideal performance indicator ring to the colour-coded tool marker (identifying the evaluated tool) that is closest to the centre of the chart, signifying the poorest performance value.

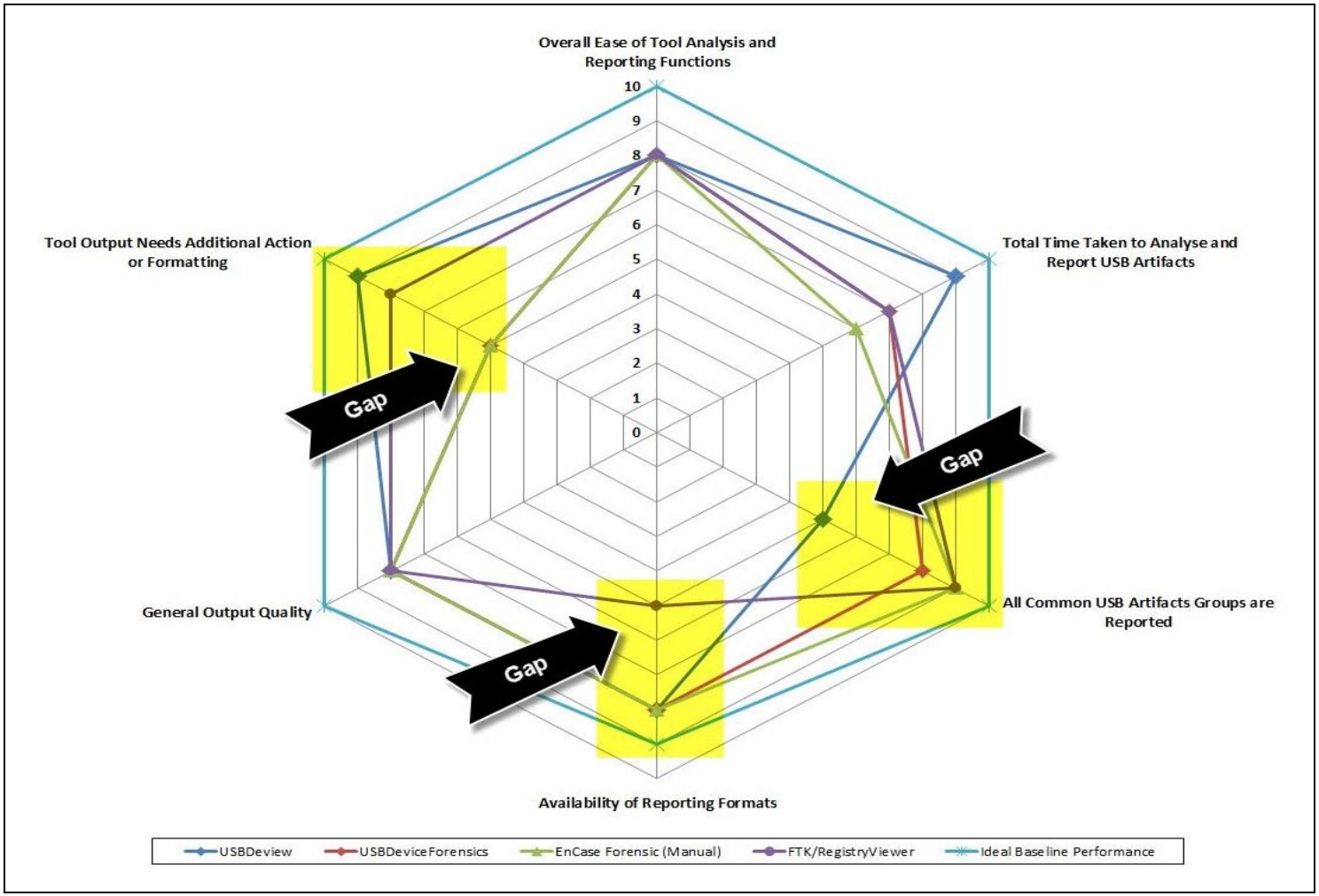


Figure 4.7. Phase Two Gap Analysis Findings

4.4. PRESENTATION OF FINDINGS – TOOLSET BENCHMARKING

The sample toolset evaluations and gap analysis results were reported in Sections 4.2 and 4.3 respectively. The toolset evaluations provided a mechanism to benchmark a sample set of commonly used USB analysis and reporting tools against each other to identify general tool capability and relative performance. The gap analysis used data from the toolset evaluations and journal entries (Refer Appendix B) to identify potential issues and areas of improvement to assist in the development of a new USB tool. The findings from the sample toolset evaluations and gap analysis are presented in Section 4.4 and are supported by visual summaries of the performance results in the form of graphs.

Figures 4.8 to 4.10 provide a breakdown of individual performance results from the manual analysis and reporting of each USB tool where applicable. Only three out of the four tool results are displayed in charts as the resulting time values were captured under manual analysis and reporting conditions. None of the tools in the sample toolset proved to have an automatic time capture function or output capability for obtaining the overall time taken to analyse and report USB artifacts.

Overall automatic tool data processing was therefore difficult to capture with an external time source during the toolset evaluations as processing of the selected data only took mere seconds to display. Figure 4.11 provides a comparison of the manual processing time results obtained for the three tools over each of the completed test scenarios. Figure 4.12 further provides a comparison between the overall tool usability of the four tools. It is based on the gap analysis results detailed in Section 4.3.2.

The USBDeview© tool performance results are presented in Figure 4.8 for overall time taken to analyse and report the various USB devices in each of the eight test scenarios. The average time taken for the USBDeview© tool in the toolset evaluations was 1 minute, 08 seconds and 89 milliseconds. The results provided the fastest processing and reporting time for all of manual evaluations. The moving trendline representation in Figure 4.8 also identified that individual analysis and reporting times became faster as the command-line extraction method was used more frequently and the researcher became more comfortable with the tool interface and specific command-line syntax being utilised.

Attention must also be drawn in the chart to the non-capture of USB device information in Test 8 as the tool can run in a Windows® 7 operating system environment but does not currently support the examination of USB 3.0 devices.

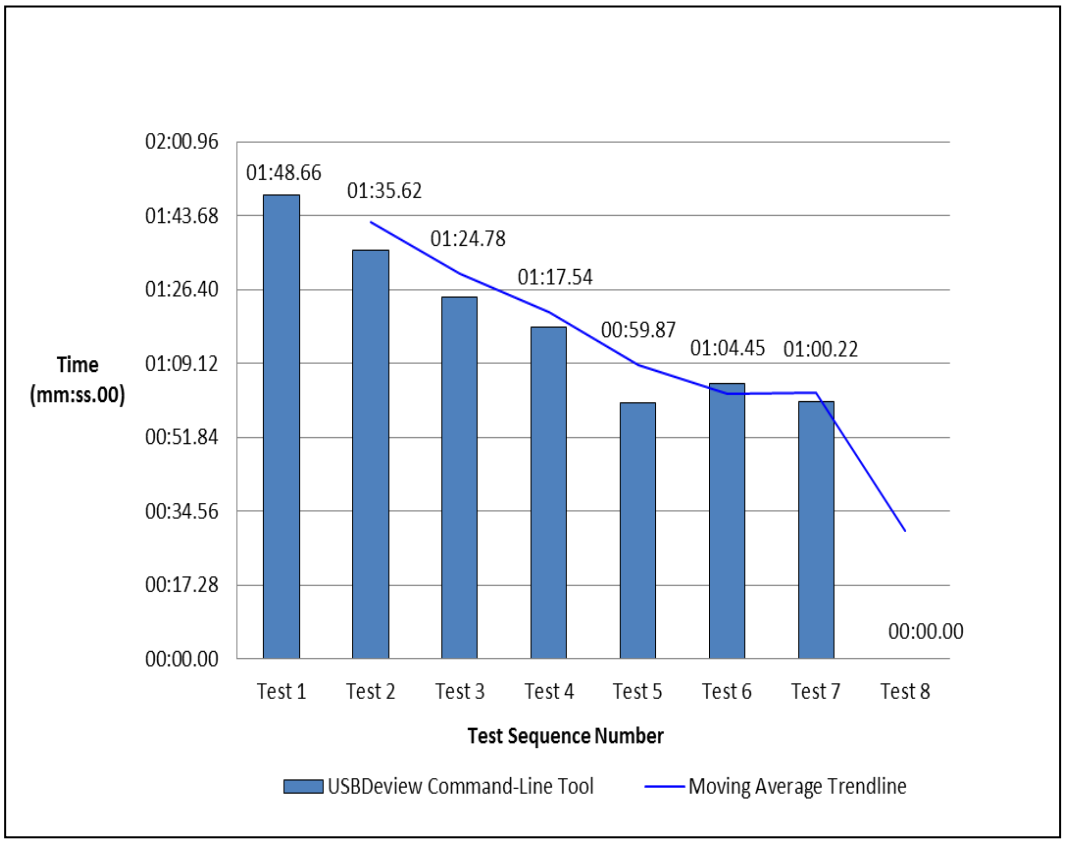


Figure 4.8. USBDeview© Manual Reporting Results

Figure 4.9 shows that the times taken for seven out of the eight individual test scenarios are very similar when the EnCase® Forensic tool was utilised. Performance hits were recorded in Test 2 as a result of the application issues discussed in Section 4.2.2.2. The extra time taken to process the artifacts in Test 2 had a flow-on effect for calculating the average time taken value across the collected time results. The average time taken for the manual EnCase® Forensic tool evaluations was found to be on the high side of 11 minutes, 0 seconds and 93 milliseconds. The resulting average time would be acceptable when analysing less than five USB devices on a single computer whilst in a forensic laboratory environment. However, it would less acceptable and more costly when dealing with many USB devices across multiple computers and sites as often is the case in large corporate investigations or onsite business environments.

The moving average trendline representation in Figure 4.9 is also affected by the issues faced in the Test 2 scenario; however, the results then flatten out for the remaining test sequences as no further technical or application issues were encountered whilst manually analysing the collected datasets with the EnCase® Forensic tool.

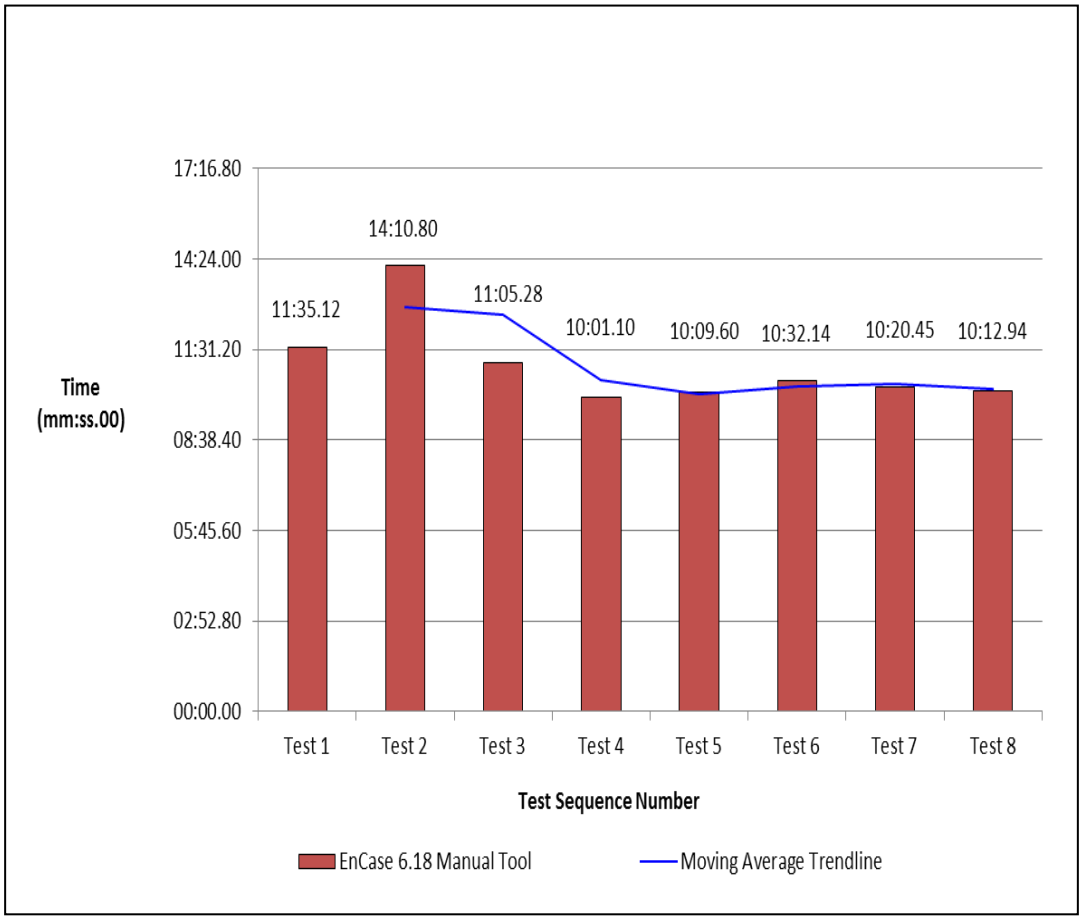


Figure 4.9. EnCase® Forensic Manual Reporting Results

Figure 4.10 shows that the time taken across the eight test scenarios for the FTK®/RegistryViewer® tool combination decreased. The moving average trend line also displays a noticeable downward pattern from Test 4 until the time values start to stabilise in the latter test scenarios. The calculated average for all of the FTK®/ RegistryViewer® tool evaluations was 3 minutes, 7 seconds and 27 milliseconds. The result is a good performance indicator for cases where multiple USB devices need to be examined in real-world examinations. In general, the FTK®/ RegistryViewer® tool combination reported the second quickest performance times across the manual evaluations shown in Figure 4.11.

The FTK®/RegistryViewer® toolset results also indicate that the use of multiple tool combinations or additional analysis functions are not likely to have a negative impact on general USB analysis and reporting performance when a trained and knowledgeable digital forensics practitioner is comfortable in using a specific USB toolset.

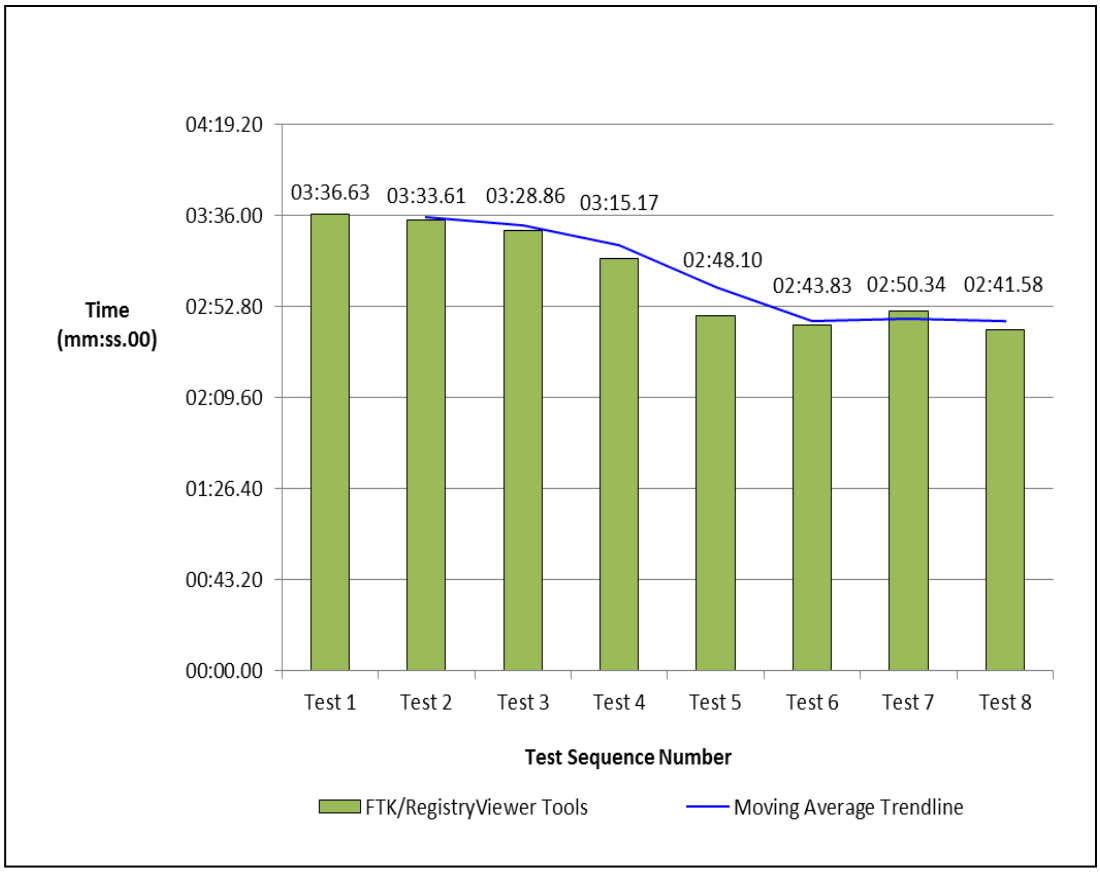
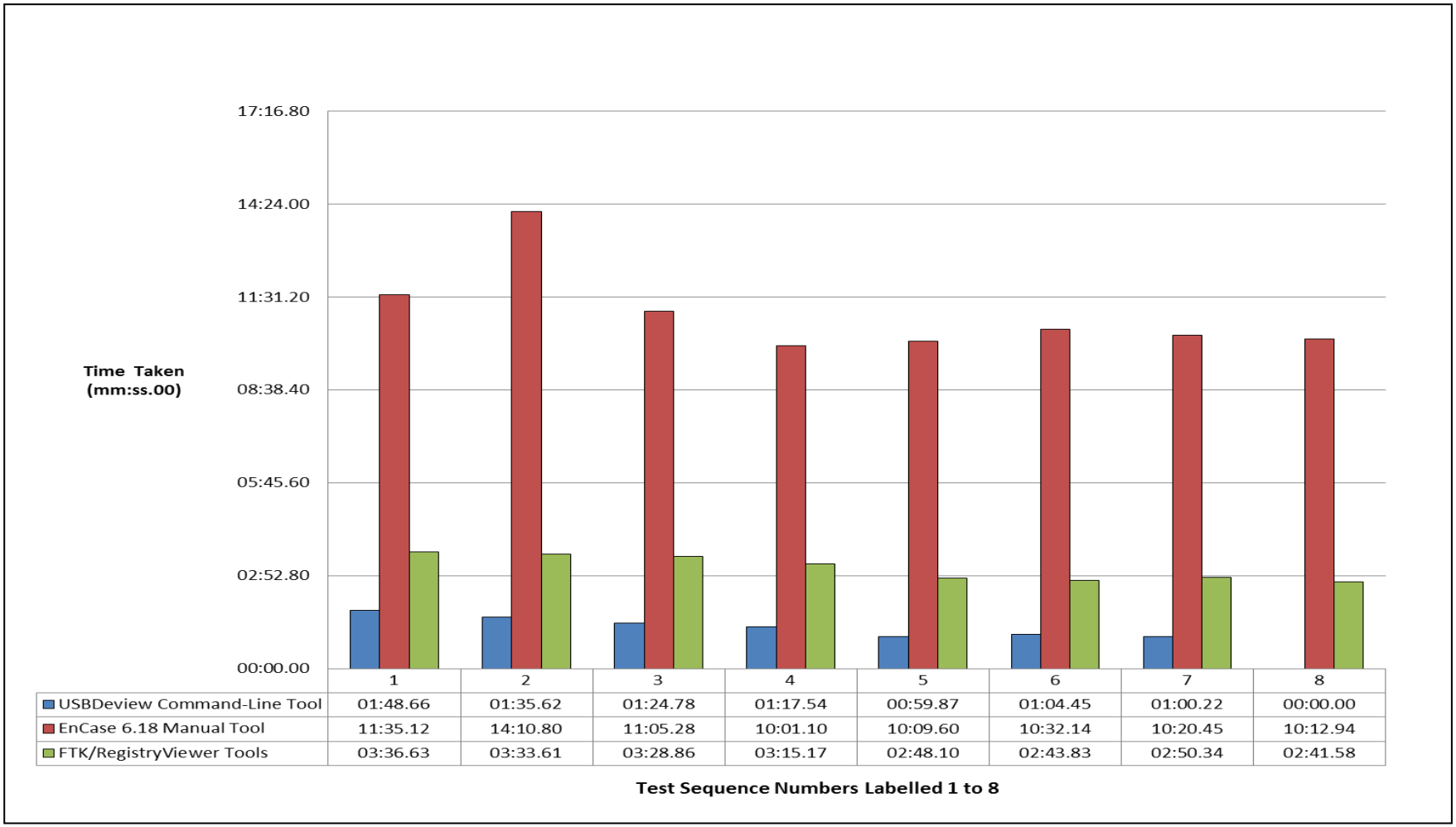


Figure 4.10. FTK® RegistryViewer® Manual Reporting Results

Timed results are presented in Figure 4.11 to give an overall performance indicator of each sample tool when compared against the other tools. The collective results are largely consistent tool by tool across the benchmark evaluations. The only notable exception is the omission of the USBDeview© tool results in Test 8 due to the tool not supporting USB 3.0 devices at the time the benchmarking occurred.

The USBDeview© and FTK®/RegistryViewer® tools that took less time would be more advantageous for digital forensic practitioners to use, when multiple devices are being examined, whilst the EnCase® Forensic tool is more suited to detailed manual analysis where there are fewer USB devices involved or when time constraints are not an issue to the case.



	1	2	3	4	5	6	7	8
USBDeview Command-Line Tool	01:48.66	01:35.62	01:24.78	01:17.54	00:59.87	01:04.45	01:00.22	00:00.00
EnCase 6.18 Manual Tool	11:35.12	14:10.80	11:05.28	10:01.10	10:09.60	10:32.14	10:20.45	10:12.94
FTK/RegistryViewer Tools	03:36.63	03:33.61	03:28.86	03:15.17	02:48.10	02:43.83	02:50.34	02:41.58

Figure 4.11. Manual USB Toolset Time Taken Comparison Results

The overall Gap Analysis results from Section 4.3.2 are presented in Figure 4.12 using a double axis column-chart. The rating values on the primary (left) vertical axis are derived from the rating system of *0* to *10* utilised in scoring the gap analysis categories for each sample tool in Table 4.6. The results identified a consistent pattern across the tool collective with each sample tool generally scoring at or considerably above the acceptable tool rating of *5*.

The secondary (right) vertical axis also shows the individual average ratings for each sample tool. Collectively, the average has been calculated at *7.50* for tool usability and functionality. The average score representation in Figure 4.12 is complimentary to the primary (left) axis data as it provides the passing reader with a quick and visual method of assessment summary without the need to further investigate each of the individual tool results.

Both the USBDeview©, FTK®/RegistryViewer® tools received the highest average score ratings (7.83 and *7.50* respectively) in the Phase Two Gap Analysis results, whilst the USBDeviceForensics© and EnCase® Forensic tools closely followed with an equal and average score rating of *7.33*. Of interest, the non-forensic USBDeview© tool scored the highest out of the three other tools for ease of use and functionality even though it was not suited for in-depth USB forensic use.

In summary, limitations in tool usability and functionality were identified across all tools during the benchmarking and evaluation processes. For example, USBDeview© was limited by design to the analysis and reporting of *SYSTEM* hive artifacts only. As a result not all the evidence groups identified in Section 3.4.2 could be captured with the tool. Data output from the USBDeviceForensics© tool and EnCase® Forensic tool (used manually without the use of automated EnScript® add-ons) required additional formatting and action by the tool user to produce usable reports. FTK® RegistryViewer® was also found to be limited by a predefined and branded HTML report format that was neither easily modifiable by the user nor able to be cleanly exported into an existing report or attachment.

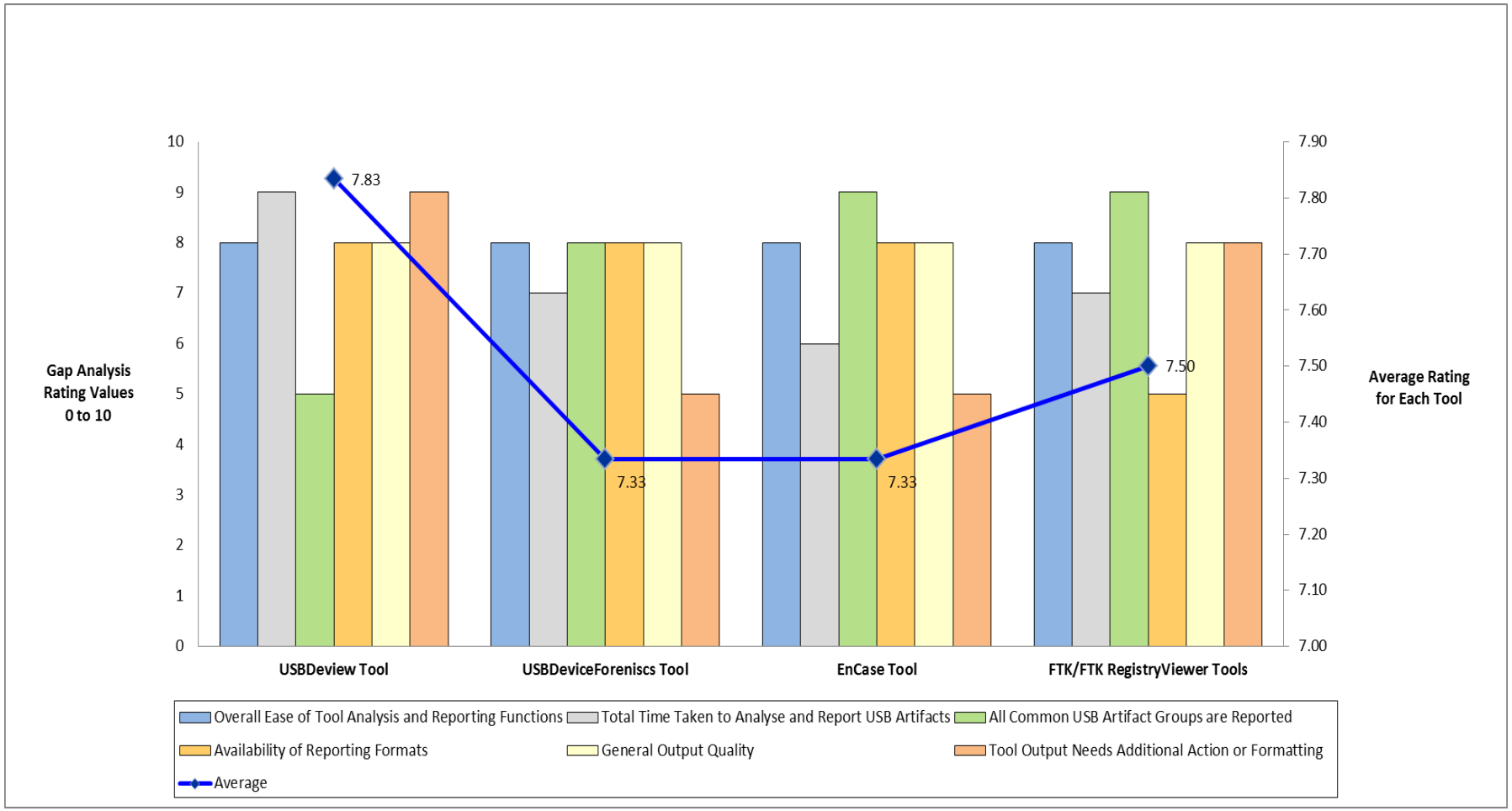


Figure 4.12. Phase Two Gap Analysis Results for the Sample Toolset

4.5. CONCLUSION

Chapter 4 reported the various research findings derived from the sample USB toolset evaluations. Benchmarking of the toolset evaluations identified similarities and limitations across the various analysis and reporting functions employed by each of the tools.

Variations to the initially proposed data requirements are outlined in Section 4.1 and then implemented in testing methodologies employed throughout the benchmarking and analysis phases. Section 4.2 provided a measure of toolset performance and capability for a sample set of commonly used USB forensic analysis tools. A multiphase gap analysis in Section 4.3 further refined the benchmarking and evaluation of the toolset, resulting in the identification of three significant limitations in tool usability, analysis and functionality. The limitations are lack of full evidence set support; unavailability of multiple reporting formats, and the need for additional data output actions on the user's part. These limitations provided the catalyst for improvements to be made in the design and capability of the new prototype tool currently in the final stages of development.

Key aspects of the toolset evaluations, datasets and gap analysis findings are summarised and visually presented in Section 4.4. Finally, comprehensive datasets collected during the evaluations and analysis phases provide a baseline for further testing, validation and presentation of findings relating to the tool development in Chapter 5.

Chapter 5

USB Tool Development

5.0 INTRODUCTION

USB toolset evaluation and software development methods were identified as an ideal combination for answering the research questions and testing the hypothesis developed in Chapter 3. Chapter 4 established a benchmark of USB toolset performance and capability by evaluating a sample USB toolset to identify limitations in their functionality and reporting outputs. The results from the toolset evaluations and associated gap analysis were then incorporated into the design process of a USB forensic analysis and reporting tool developed as part of the current research.

Chapter 5 reports on the development, validation and testing of a prototype USB forensic tool, identified hereafter as the USBForensicReporter© tool. Section 5.1 discusses the design, operation and reporting functionality of the tool. Section 5.2 discusses previous research relating to examining and reporting of raw Registry files. Earlier research was important to understanding the inner workings of a Windows® Registry from a development aspect as core-components are not well documented in the public arena by Microsoft®. The section concludes by providing validation and field testing results. Section 5.3 reports on field testing of the USBForensicReporter© tool against the toolset benchmark results from Chapter 4.

Finally, Section 5.4 presents and explains major research findings of specific performance and functionality indicators collected during the earlier sample toolset evaluations and field testing of the newly-developed USBForensicReporter© tool. The findings in Chapters 4 and 5 will also provide the foundation for final interpretation and discussion of the research results in Chapter 6, in-order to answer the research questions and test the hypothesis statement.

5.1. TOOL DEVELOPMENT

The development of the USBForensicReporter© tool focused on three main areas relating to overall design, tool operation and data output. Section 5.1.1 discusses the tool's frontend GUI component for user interaction and the backend processing component for data extraction, analysis and reporting purposes. Section 5.1.2 provides a general overview of tool operation akin to providing the digital forensic practitioner with a walk-through of the tool's use and functionality. Section 5.1.3 identifies the tool's generation of data output into usable reporting formats.

5.1.1. Tool Design and Alterations

The tool was created using Microsoft® Visual Studio® 2010 Professional (version 10.0.30319.1) and written in the Visual C++® programming language. A number of alterations to the proposed software design process (as outlined in Section 3.3.3) were implemented in the design phases due to several technical issues. The issues were encountered when developing a workable mechanism to mount the forensic image files in the tool interface. Coding issues were also encountered that resulted in the USB evidence groups identified in Table 3.2 not being fully parsed out and reported on from each of the raw Registry files.

The final design architecture was redesigned to incorporate a methodology for comparison between "offline" Registry Files (extracted from the suspect evidence files) and Windows® Registry files from a "live" computer system. The overall design process was separated into two modular components: developing the back-end processor (or engine-room) of the tool and developing the front-end GUI for user interaction. The separate design stages simplified the tool development and allowed testing and debugging actions to be conducted on sample data taken from the earlier sample USB toolset evaluations in Chapter 4.

The back-end processing module was developed to read both the "offline" suspect and "live" Registry files (namely the *SYSTEM*, *SOFTWARE*, *NTUSER.DAT* files, and the *setupapi.dev.log* files for each user account) along with the physical USB suspect device.

Registry, HTML, MD5 Hashing and standard function call modules were individually coded and utilised during the main processing routines so as to capture the various keys, subkeys and data values associated with the selected Registry hive files.

Section 3.1.3 identified previous research from DeAbren (2000) relating to the implementation of test functions and related software within the design of tool modules. Comparable elements and methods were implemented in the design and coding of the new USBForensicReporter© prototype tool. Testing routines, step-through-analysis and debugging methods were utilised in the programming coding to ensure that the correct data strings were being parsed by the tool and to provide quality data output for the end-user.

All function calls, identified keys, subkeys, and string data values are recorded in the form of a text or log file each time the tool is run. Key data blocks are also written to an HTML report template during processing. The HTML report provides a focal-point of core forensic artifact information relating to the suspect Registry files and physical USB device for the tool user and intended recipient to examine.

The tool's GUI interface module was designed for ease-of-use with only the need for minimal case and USB device information to be entered by a user in the top information bar area of the tool interface. Two program buttons, one to process the evidence set and one to exit the program, are further provided in the bottom bar area to minimise additional actions on the user's part. This makes the tool interface very simple to use and keeps the workload of the user to a minimum. The development of the GUI usability is therefore aimed at any level of user's expertise from trainee to skilled practitioner.

5.1.2. Tool Operation

In its current prototype version (1.0.6) the USBForensicReporter© tool is based on the "analyst workstation" concept. That is, analysis and reporting of artifacts are completed by the digital forensics practitioner at an analyst workstation level in a laboratory environment. The tool is specifically designed to conduct both comparative analysis and reporting on a suspect USB storage device and Windows® Registry files obtained from a forensic image copy of a suspects hard drive.

Where there is no suspect USB storage device available for examination, the tool will still be able to undertake a forensic analysis of the Registry files from the suspect computer system for reporting of previously connected USB devices only.

Under typical examination conditions the practitioner will apply write-blocking technology (adhering to established local operating procedures or forensic best practise so as to prevent data loss or alteration of USB device content). The suspect USB storage device will then be connected to either USB 2.0 or USB 3.0 supported ports on the workstation. The suspect device can then be disconnected as this user action only seeds the Registry and system files of the "live" Windows® 7 workstation with USB data pertaining to the suspect device.

The complied tool is launched from a self-contained Windows® executable file (comprising of the required DLLs and Visual® C runtime libraries) at any location on an analyst workstation whilst utilising elevated Windows® 7 Administrator User Account Control (UAC) privileges. The user enters the relevant case or matter information before selecting a set of source evidence Registry files from a dropdown dialog box. The evidence source consists of the relevant *SYSTEM*, *SOFTWARE*, *NTUSER.DAT* and *setupapi.dev.log* files that had earlier been extracted from the mounted suspect evidence file. The evidence source data is loaded into the analyst workstation's "live" Registry as independent and read-only Registry files utilising the *Load Hive* function of the Windows® Registry Editor (Regedit) when the *Process Evidence* button is selected by a user. The *setupapi.dev.log* file must be in the same directory as the other Registry file for data to be captured.

The data analysis and comparison process flow is then initiated by the user as depicted in the proposed tool flowchart of Table 3.4. The processing features of the tool are unique and innovative when compared to existing USB analysis and reporting tools such as those in the sample toolset. The USBForensicReporter© prototype tool conducts a comparative analysis of the suspect Registry files and the "live" system Registry files to make a determination if the physical USB device has been previously connected to the suspect computer system. None of the sample tools used in the toolset evaluations have this type of processing functionality.

The sample tools each rely on either "offline" analysis of evidence files, individual Registry files, or "live" USB device analysis of a connected device. Figure 5.1 provides a graphical representation of the tool operation with the case information having been entered by the user and tool output from the suspect USB device, suspect Registry files and "live" forensic workstation Registry files being displayed to the user via the output section of the tool interface.

The output in the GUI interface presents the user with different dates and time stamps associated to the current analysis session and also to the various Last Written date and time stamps extracted from the registry keys as each suspect hive file is read line by line. All current processing time stamps rely on the system clock of the analyst workstation being checked or correctly configured to an external and reputable time source before the tool operation is commenced.

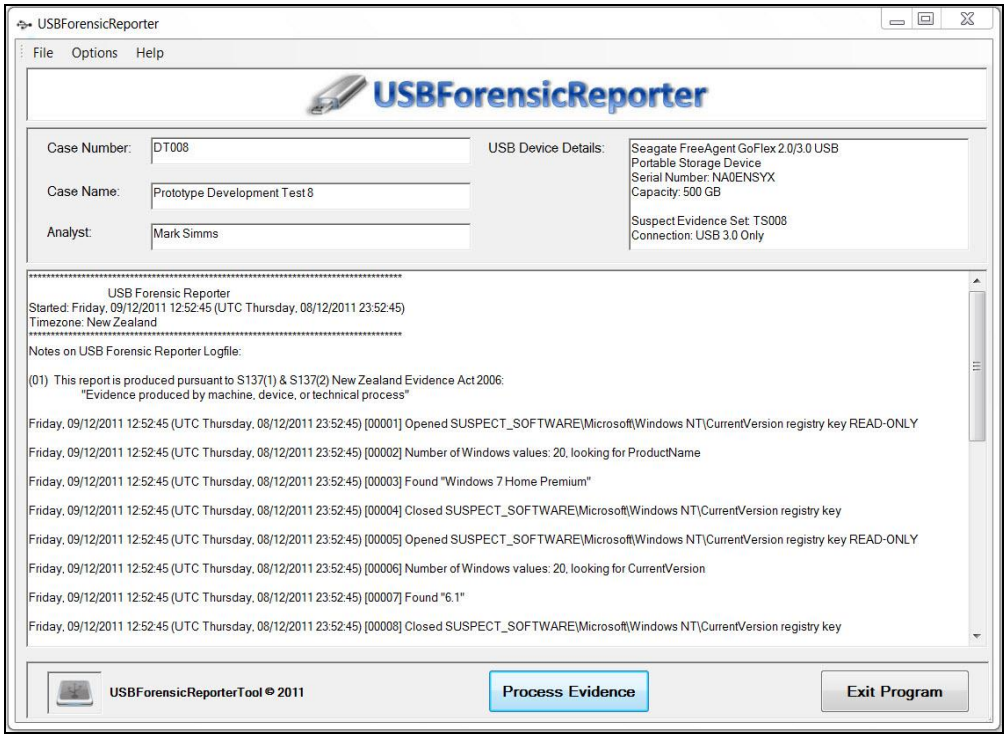
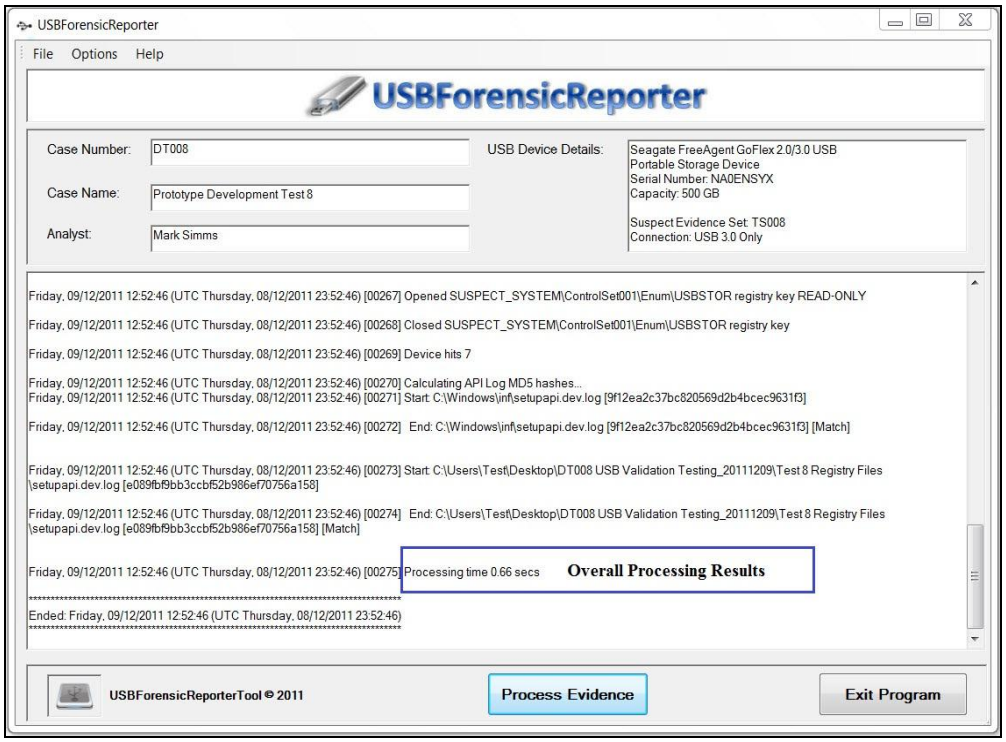


Figure 5.1. USB Prototype Tool Interface and Processing Output

Figure 5.2 shows the final results of analysis phase as presented to the user of the tool. The resulting screen output is progressively translated into both log and HTML file output at the start, during and at the completion of the processing action. The overall processing time results is one of the last entries to be recorded and presented by the tool and is highlighted in Figure 5.2.

The actual processing time is derived from a clock routine that uses the calculation of "end time – beginning time /1000 (Clock Ticks per Seconds) = return time in seconds and or milliseconds". Obtaining an actual processing time value is an important function of tool logging for benchmarking automated analysis and reporting toolsets in order to determine the overall performance indicators of a tool. None of the toolset evaluation examples had this type of functionality in their current tool designs. Logging of processing time was therefore another important feature to be added in the design phase of the prototype tool.



F*igure 5.2.* Completed USB Prototype Tool Operation with the Processing Results Highlighted

Reporting of data output from the prototype tool is discussed in more detail in Section 5.1.3.

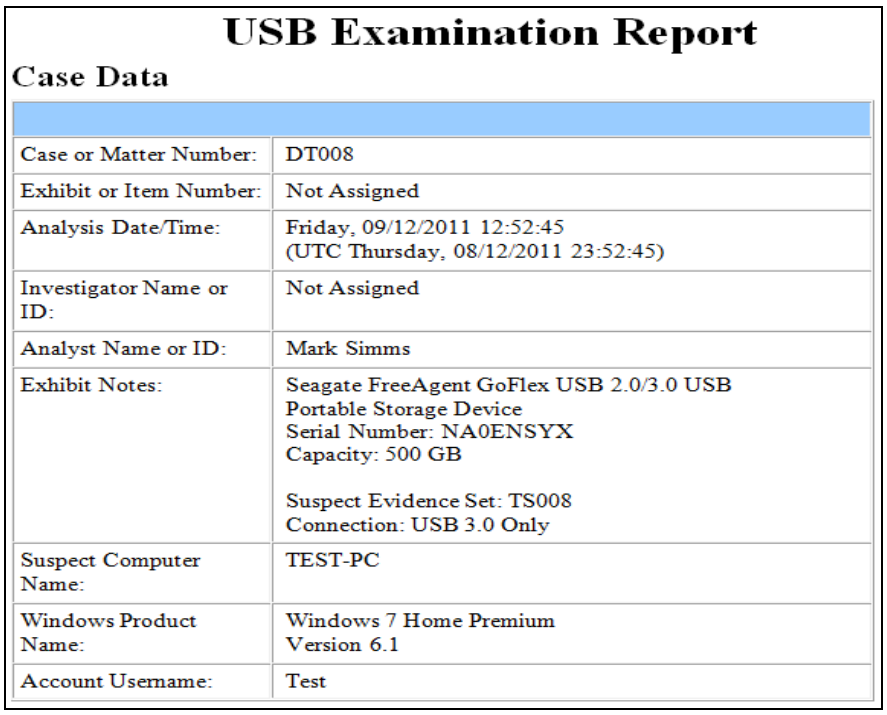
5.1.3. Data Reporting

Phase Two of the gap analysis in Chapter 4.3.3 identified the need for more than one reporting format to be provided by USB analysis and reporting tools. Both HTML and text-based log file reporting formats were implemented in the current tool development using the proposed CLC/RAD design framework model depicted in Figure 3.3 and the flowchart presented in Figure 3.4.

The USBForensicReporter© tool will automatically generate both reporting output formats each time Registry and system files are processed. The subsequent HTML report contains a summary of key USB device information resulting from past connection of physical USB devices on the suspect computer system and recent connection to the analyst workstation.

The HTML report is divided into two main sections. The first part contains the case and device information entered by the user into the tool GUI and the second part contains details of all USB storage devices that have previously been recorded in the suspect Registry files. The "Case Data" section in Figure 5.3 presents the first part of the report. It provides "chain of custody" information in the form of USB device, investigator and analyst or user details pertaining to a single USB device examination. Cosic and Baca (2010) identified that "chain of custody" is a vital component in an evidence life cycle.

Procedures and records relating to forensic examinations must be auditable and well documented for digital evidence to be ultimately accepted in legal proceedings. Each field in the report is adaptable for use in both law enforcement and corporate environments alike. Additional scene or laboratory notes can also be added into the expandable USB Device Details section to provide a complete record of the device examination in question.



USB Examination Report

Case Data

Case or Matter Number:	DT008
Exhibit or Item Number:	Not Assigned
Analysis Date/Time:	Friday, 09/12/2011 12:52:45 (UTC Thursday, 08/12/2011 23:52:45)
Investigator Name or ID:	Not Assigned
Analyst Name or ID:	Mark Simms
Exhibit Notes:	Seagate FreeAgent GoFlex USB 2.0/3.0 USB Portable Storage Device Serial Number: NA0ENSYX Capacity: 500 GB Suspect Evidence Set: TS008 Connection: USB 3.0 Only
Suspect Computer Name:	TEST-PC
Windows Product Name:	Windows 7 Home Premium Version 6.1
Account Username:	Test

Figure 5.3. HTML Report Extract Containing Case Data

The main body of the HTML report is reserved for reporting of all USB devices that have previously been attached to the computer system under examination. The *USB Devices Details* section makes use of colour-coded panels located at the top of every subsection for each of the USB devices being reported. A *blue* panel represents reporting of device details from the suspect Registry only, whilst a red panel represents the flagging of a *Device Alert Status* to the user. The *Device Alert Status* is unique to the USBForensicReporter© tool and assists a practitioner with prioritising analysis needs when dealing with multiple USB storage devices.

The triggering of a *Device Alert Status* flag during the tool processing and reporting phases is based on comparison matches being identified between the suspect Registry files and "live" analyst workstation Registry files. Key values such as the *device descriptor's iSerialNumber* (Identified in Figure 4.3) are central to the success of the comparison analysis method being employed in the tool design. Figure 5.4 shows a reported result of **No matches** in the *Device Alert Status* field, indicating that the suspect USB device had not previously been connected to the suspect computer system.

Device Number:	2
VendorID:	058F
Product ID:	6377
Version:	1.03
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB MS Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&3
Mounted Devices: Last Drive Letter Mapping:	H:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
GUID Number:	{3d03487e-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:20.644 NZST
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 NZST (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	H:\ Friday, 05/08/2011 17:34:00 NZST (UTC Friday, 05/08/2011 05:34:00)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 NZST (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Figure 5.4. HTML Report Shows "No Matches" in the Device Alert Status Field

On the other hand, Figure 5.5 shows a reported **Alert: Device Found** result in the *Device Alert Status* field indicating that the suspect USB storage device had previously been connected to the suspect computer system.

Device Number:	7
VendorID:	0BC2
Product ID:	5031
Version:	_210
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Seagate FreeAgent GoFlex USB Device
Class:	DiskDrive
Device Serial Number:	NA0ENSYX&0
Mounted Devices: Last Drive Letter Mapping and or Disk Signature Values:	F:\ 30 05 54 C5
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{078ccf70-c6f3-11e0-bc58-001fd0060cfd}
User Account NTUSER.DAT: MountPoints2:	Device has been used by the user account: Test
Setupapi.dev.log: First Device Connection Date/Time:	15/08/2011 16:08:05.489 NZST
Device Classes: First Connection Date/Time After Reboot:	Monday, 15/08/2011 16:08:06 NZST (UTC Monday, 15/08/2011 04:08:06)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	No entry for external USB drive
USBSTOR: Last Written Date/Time Stamp:	Monday, 15/08/2011 16:08:06 NZST (UTC Monday, 15/08/2011 04:08:06)
Device Alert Status:	ALERT: DEVICE FOUND

Figure 5.5. HTML Report Shows "Alert Device Found" in the Device Alert Status Field

The resulting text-based log file has been developed to be used as an integral part of the digital forensic practitioner's case and analysis notes for the USB storage device examination. The log file provides a printed record of the tool's process, data extraction, analysis and reporting activities to assist in the exemplification of examination findings. Log file output and case notes are also useful for peer-reviews of analysis and reporting findings and for further investigation or disclosure requirements at a later stage.

International and local best forensic practice also recommend that analysis actions and data results should be thoroughly documented by the practitioner as full disclosure of such files may be a legal requirement in the event of criminal or civil proceedings.

Both reporting formats produced by the USBForensicReporter© tool provide a provision for any resulting data output to be produced under New Zealand legalisation, and within the current provisions of the Evidence Act 2006 (New Zealand Government, 2010, Evidence Act 2006, s137). The legislative enactment in this case allows electronic forms of evidence to be produced in a court of law (whether totally or partially obtained) from a "machine, device or technical process", (Subsection 1, para.1).

Key areas of a legal challenge to electronic evidence of this kind centre on the operability and reliability of the machine, device or technical process producing the evidence. The principle of common law presumption acts as the test for machine produced evidence. In short, there must be a belief on the Court's part that under ordinarily or usual circumstances the computer-based tool (in the case of forensic software tools) produced the evidential data in a normal manner as it was intended to do so for evidence to be accepted, unless opposing evidence is produced to the contrary (Harvey & Ayers, 2010).

Whilst the USBForensicReporter© tool is a prototype design at this stage of the research there is likelihood that the tool could be used in future real-world USB forensic examinations once extensive data validation and tool enhancements have taken place. With that in mind, all reporting templates used by the tool to produce computer-based evidence output have the wording "This report is produced pursuant to S137 (1) & S137 (2) New Zealand Evidence Act 2006: Evidence produced by machine, device, or technical process" incorporated into their design in order to conform to prescribed legal requirements in this country.

5.2. TOOL VALIDATION

In order for a USB analysis and reporting tool to be accepted into forensic laboratory environment, tool testing validation and verification of data output must be conducted by the practitioner before it is used for actual case evidence. Section 3.3.3 previously discussed established tool testing programs and validation guidelines by international organisations such as NIST (2005) and SWGDE (2006). NIST provides test results for a wide range of forensic software and hardware types, yet no specific USB analysis tools have been validated to date under the CFTT program.

Likewise, software vendors have largely not made tool testing and validation results publically available to test their findings or to determine if the tool does what the vendor claims it is capable of doing. Therefore, in this research the adopted tool validation methodology uses previous research elements from a representative sample of three credible and published researchers (Thomassen, 2008; Norris, 2009 and Carvey, 2011) to assist in the tool validation process. Tool data verification was also carried out using the benchmarked test results from Test 1 in Section 4.2.2.1 and the EnCase® tool software for output verification.

5.2.1. Previous Research – Registry Structure Discovery

There are still some gaps in understanding the Windows® Registry at both a scientific and industry level. These knowledge gaps are due in part to Microsoft® not making the complete binary structure of the individual hive files publically available in any documented detail (Microsoft, 2011b).

Recent research has however provided great inroads into forming a greater understanding of forensic aspects related to the internal structure of files associated to the Windows® Registry. Carvey (2011) provided a clear and detailed understanding of the registry binary structure using different levels of analysis. Important signature, key and cell values were identified using both hexadecimal values and descriptions that would be beneficial to digital forensic practitioners and researchers alike when examining registry files at a binary level.

Whilst Carvey's latest research (Carvey, 2011) was published after the current prototype tool design was started, the specific key and cell values have been hugely beneficial in providing relevant references when debugging tool code and troubleshooting output issues during the later stages of the SDLC cycle. Internal Registry structures have also been identified and published in previous research by Thomassen (2008) and Norris (2009). Windows® Registry hive files typically consist of a header signature (identified as *regf*) and *Base Block* information, with additional sections that are defined by each of the researchers as *bin* files.

The *bin* files are identified by an *hbin* signature and consist of different cells containing various key and value data strings for a length of 4096 bytes per block.

Each of the previous research examples are invaluable forensic resources to the digital forensics practitioner who is tasked with examining both current and deleted Windows® Registry hive files. Likewise, the Windows® Registry structure has remained fairly stable and similar in appearance between releases of the Windows Vista® and Windows® 7 operating system versions. Therefore, common signatures and cell values will be more easily recognisable when a practitioner examines associated binary files and unallocated space with a hex editor or common forensic software tools such as EnCase® Forensic and FTK®.

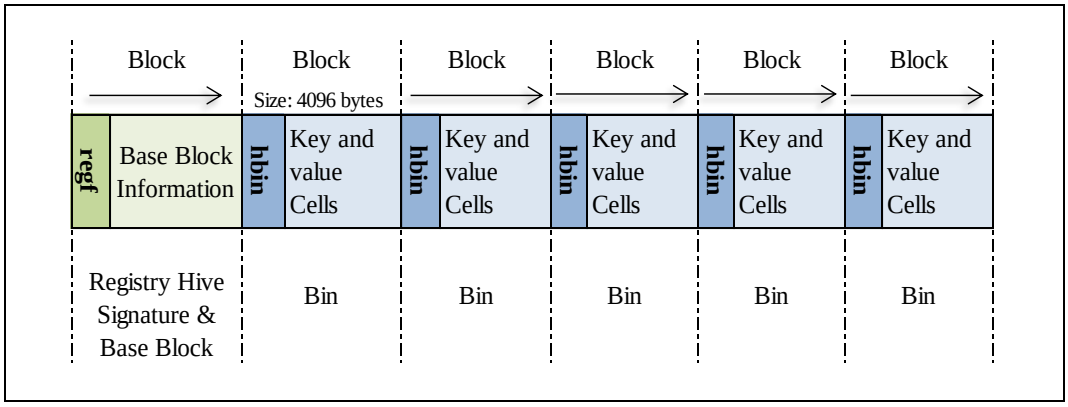


Figure 5.6. The Windows® Registry Hive Structure. Adapted with permission from Thomassen (2008, p.7)

5.2.2. Tool Related Registry and Forensic Artifact Validation

The term *Validation* is defined by the IEEE Standards Association (IEEE, 2005) as follows:

> **(A)** The process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements. **(B)** The process of providing evidence that the software and its associated products satisfy system requirements allocated to software at the end of each life cycle activity, solve the right problem (e.g., correctly model physical laws, implement business rules, use the proper system assumptions), and satisfy intended use and user needs. (p.9)

The validation concept is one of the components of the widely accepted IT industry software Verification and Variation (VV) process model.

The VV model is used by software engineers and designers to implement quality assurance processes in all facets of the software lifecycle. Similar concepts have been used by other researchers. A review of the research studies from Section 5.2.1 found that Thomassen (2008) had successfully validated field data output contained within deleted key cells from sample Registry hives used during her testing.

A similar validation methodology is also applicable for "offline" Registry files that are examinable in both binary and mounted file formats. Validation of Registry key and value data can therefore be achieved in the current research by using known hive files (i.e. the Registry files from Test 1 in a raw, non-mounted state as referred to in Section 4.2.2.1) to determine if specific keys and cell information are being parsed correctly or not by the processing actions of the USBForensicReporter© tool.

The proposed tool validation will use a value validation method encompassing both analysis and testing of the relevant *SYSTEM, SOFTWARE, NTUSER.DAT* and *setupapi.dev.log* files. Programming routines, data output and the mounting of suspect Registry files on a Windows® 7 operating system were also examined at different intervals during each of the SDLC phases to further assist in validation and debugging of the program code.

Figures 5.7 to 5.10 are representative samples of the *value validation* method employed throughout the tool validation process. The sample is taken from the collected *SYSTEM* hive data in Test 1 of the toolset evaluations. The *SYSTEM* hive contains the most relevant number of USB artifacts for a digital forensic practitioner to review in the first instance of an examination and is therefore an ideal candidate for validation purposes.

Only fundamental Registry key and cell values that are relevant to digital forensic practitioners have been deconstructed and validated in the screenshots of each figure. Where applicable hexadecimal values have also been represented in little-endian ordering for each figure. That is, the Most Significant Byte (MSB) in a data sequence starts on the right and the Least Significant Byte (LSB) is on the left (Intel, 2004; Casey, 2011).

Figure 5.7 displays the *Base Block* of the *SYSTEM* hive using the WinHex® binary or hex editor software. The *Base Block* provides the hive signature of "*regf*", last written date and time stamp values, and file name information.

The decoded last written time stamp value of *05 August 2011 21:26:42 (+12:00 UTC)* is an important artifact when constructing timelines relating to previous USB connection activity on a computer system.

According to Microsoft® (2011c) the 64-bit hexadecimal timestamp value "represents the number of 100-nanosecond intervals that have elapsed since 12:00 a.m. January 1, 1601 Coordinated Universal Time (UTC)". In a Windows® 7 Registry, the last modified (or written) 64-bit date and time stamp is only found in registry keys, and not in value cells. The highlighted details derived from the *Base Block* were successfully validated against the binary file and references produced by the three researchers (Thomassen, 2008; Norris, 2009 & Carvey, 2011).

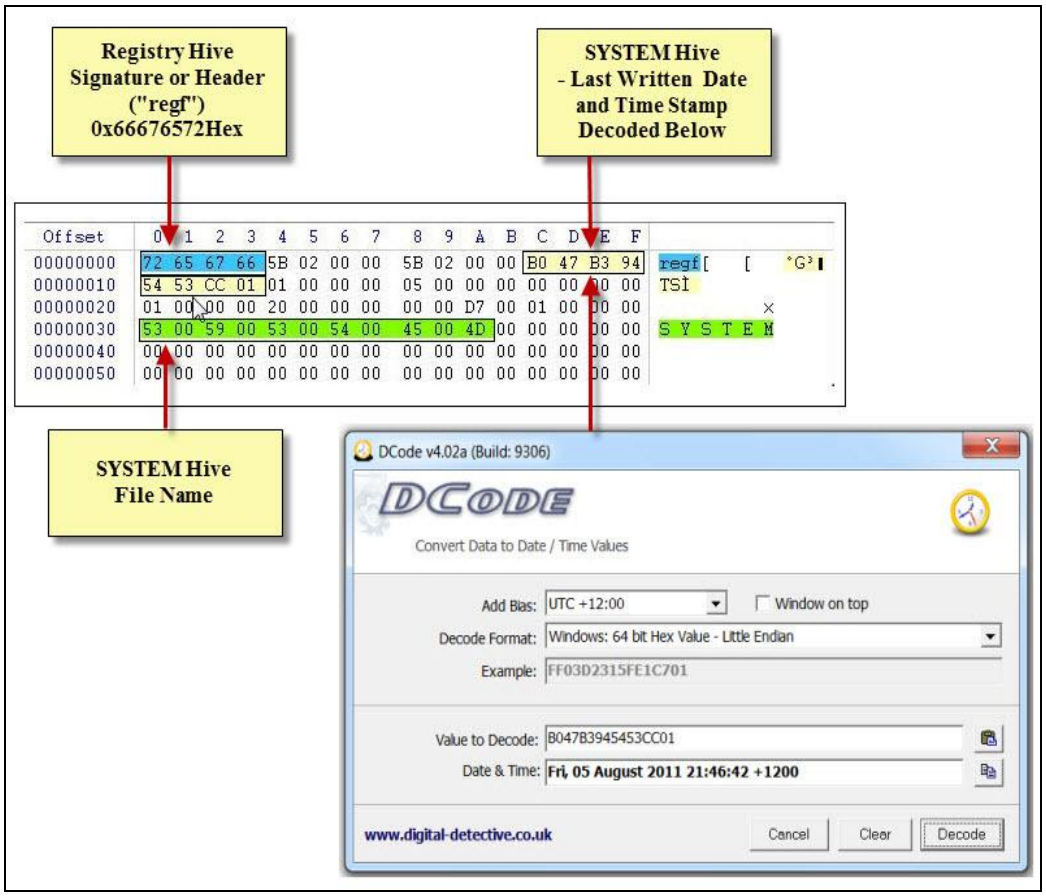


Figure 5.7. Hive Base Block Extract Details

Figure 5.8 displays extracts from the *Hive Bin* (also known as *Hbin* or simply *bin*). Values of interest that were deconstructed and validated include the *hbin* signature and the bin size that throughout the validation process was found to be consistent size of 4096 bytes for each bin example examined.

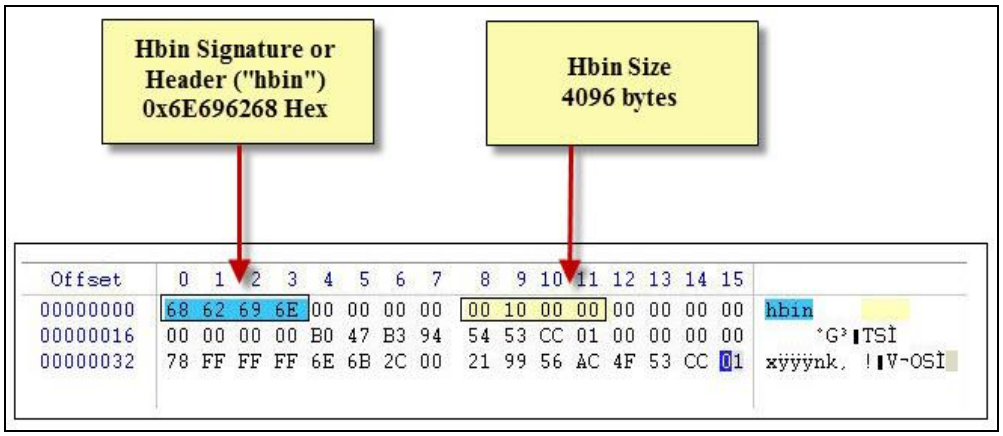


Figure 5.8. Hbin Extract Details

Figure 5.9 displays the most critical information resulting from a binary examination of the *USBSTOR* subkey location of the *SYSTEM* hive. The *device descriptor iSerialN*umber (Refer Figure 4.3) of the SanDisk Cruzer USB device was validated along with Last Written timestamp that is stored as a 64-bit hexadecimal value at byte offset *0x08* for 8 bytes.

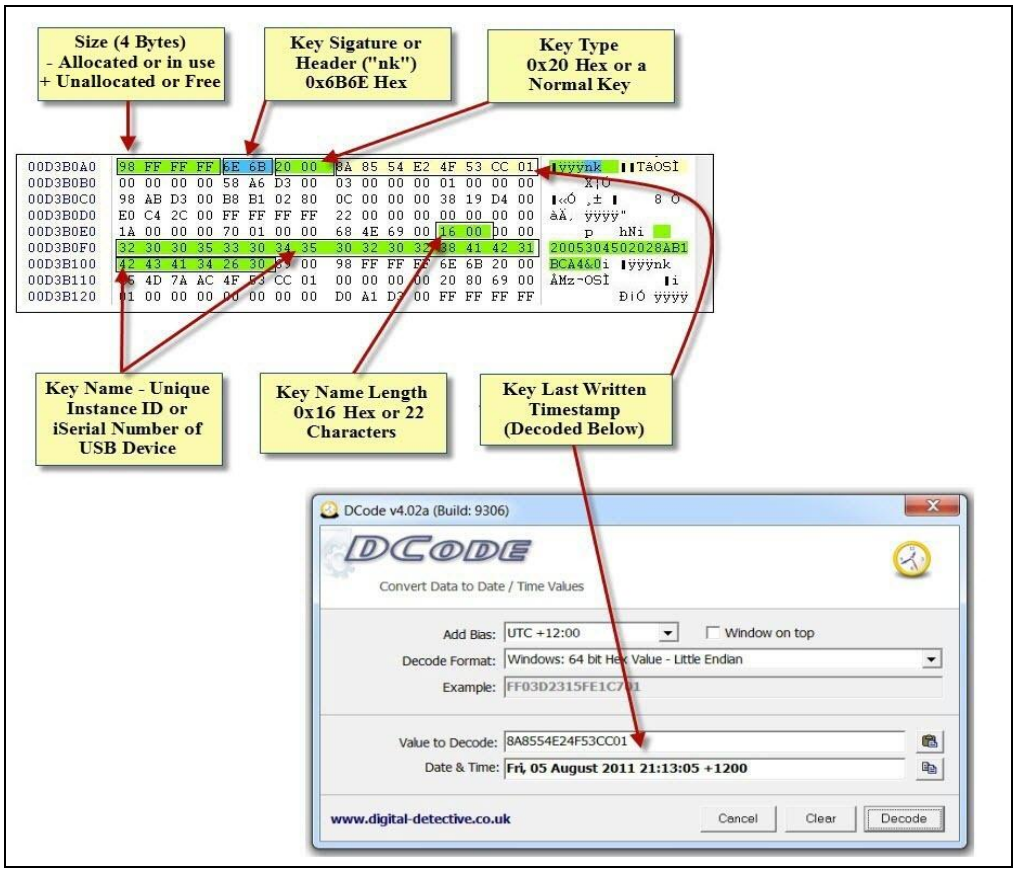


Figure 5.9. Key Cell Extract Details

Size values are interpreted as integer values with current keys and values having a negative size, and deleted keys or values having a positive size. The key cell signature or header in Figure 5.9 is displayed as "*nk*" and according to Harvey (2011) the next hexadecimal "key node type of 0x20 represents a normal key" as opposed to a Root key with a "node type of 0x2C" (p.28) such as the *SYSTEM* hive.

Figure 5.10 deconstructs the *FriendlyName* and string values associated to a SanDisk Cruzer 4GB USB device used in the Test 1 sample toolset evaluations. Device friendly names along with *UIID* values (such as the *iSerial* number) offer the digital forensic practitioner the best sources for keyword or registry searching when conducting Windows® related USB examinations.

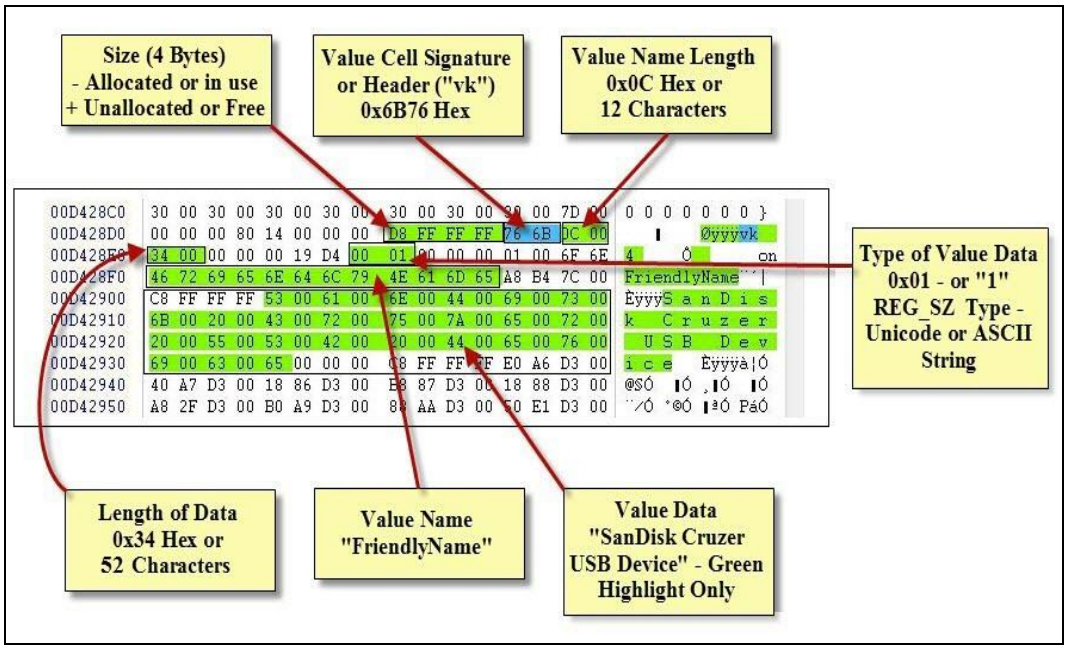


Figure 5.10. Value Cell Extract Details

The value signature of "*vk*" (blue highlight) and the other green highlighted data extracts were successfully validated using a combination of the *value validation* method, the binary hive file and hive references produced by Thomassen (2008) and Norris (2009).

5.2.3. Data Verification Method

Verification is the final component of the VV process model previously identified in Section 5.2.2. The term *Verification* is also defined by the IEEE Standards Association in similar terms as the IEEE V*alidation* definition in Section 5.2.2.

Verification is defined by IEEE (2005) as follows:

> **(A)** The process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase. **(B)** The process of providing objective evidence that the software and its associated products conform to requirements (e.g., for correctness, completeness, consistency, accuracy) for all life cycle activities during each life cycle process (acquisition, supply, development, operation, and maintenance); satisfy standards, practices, and conventions during life cycle processes; and successfully complete each life cycle activity and satisfy all the criteria for initiating succeeding life cycle activities (e.g., building the software correctly). (p.9)

Guo, Slay and Beckett (2009) best describe verification in the context of digital forensic examinations as the "confirmation of a validation with laboratories tools, techniques and procedures" (p.S13). Established laboratory tools and analysis techniques were employed in the current research to verify data output from the USBForensicReporter tool using VV concepts also identified by Fisher (2007). The data verification phase provided conformation of the previous binary validation and compared output report results from the developed tool against another benchmarked USB tool (EnCase® Forensic) which is commonly used across multiple agencies and corporate digital forensic entities in New Zealand.

The test case used in this phase was named *DT001* and contained the SanDisk Cruzer USB device and suspect registry data from the validation test in Section 5.2.2. This evidence set was a consistent and known source of evidential data to assist in producing the verification results. Table 5.1 displays a summary of the data output verification results. Reported USB artifacts from both output reports of the USBForensicReporter© and the *EnCase*® Forensic tools were compared against each other in relation to data details shown in the Table 5.1, and in more specific detail in Appendix E.

The output reports from the USBForensicReporter© prototype tool were found to be accurate in both USB forensic artifact detail and reported timestamp values. Therefore, the prototype tool was found to operate in the prescribed manner and to report reliable data output from the suspect Registry files.

Table 5.1

Test Case Output Verification Results

USB Artifact Verification Details	EnCase Forensic (v6.18) Commercial Forensic Tool	USBForensicReporter Tool (v1.0.6) Prototype
Vendor, Product and Version Identification Details	✓	✓
FriendlyName	✓	✓
Device *iSerialNumber*	✓	✓
USBSTOR Last Written Timestamp	✓	✓
MountedDevices Mapping	✓	✓
Unique Instant ID (Encompassing the Device iSerialNumber)	✓	✓
DeviceClasses GUID Identification and Last Written Timestamp	✓	✓
NTUSER.DAT – MountPoints2 User Account Identification	✓	✓
Windows Portable Devices Mapping and Last Written Timestamp	✓	✓
Seupapi.dev.log – 1st Device Connection Details and Timestamp	✓	✓

Achieving a successful validation and verification outcome allowed the tool to be field tested against previous toolset evaluation results in Section 5.3.

5.3. PROTOTYPE FIELD TESTING AND DATA ANALYSIS

The conditional requirements evaluation criteria and sample toolset evaluation results from Section 4.2 were used as a baseline of measurement and analysis during field testing of the USBForensicReporter© prototype tool. Field testing enabled both performance and functionality to be accessed, and data output reports to be generated for analysis against an established and previously benchmarked dataset.

The same six USB related memory and PSD devices from the earlier toolset evaluations were connected and then disconnected one at a time to a Windows® 7 analysis workstation in the laboratory.

The suspect Registry files from the earlier benchmark evaluations were also loaded into the Windows® Registry of the "live" analysis workstation as independent and read-only Registry files for comparison purposes by the prototype tool.

The latest version of the USBForensicReporter© prototype tool (version 1.0.6) was used for each sequence run of the testing process. Test sequences labelled as DT001 to DT006 (i.e. Development Test###) established the extraction, analysis and reporting functions of the new prototype tool using USB 2.0 supported thumb drive and PSD devices. A further test labelled as DT008 was also conducted to confirm the tool's functionality whilst using a single USB 3.0 supported PSD device. Each test sequence was run without any technical issues being encountered by the developed tool.

Table 5.2 presents the individual processing times for each of the seven field test sequences that were conducted. Within the test results, Test DT006 was noticeably slower for no apparent reason when compared to the other test results. Ultimately the developed tool's processing, analysis and reporting performance results will always be governed by type of analyst workstation being utilised, and the amount of USB devices that had previously been connected to the suspect computer system.

Table 5.2

USBForensicReporter© Field Test Results

Test Sequence Number	Development Test Processing Results
DT001	00:01.62
DT002	00:01.10
DT003	00:01.24
DT004	00:01.31
DT005	00:01.45
DT006	00.01.75
DT008	00:00.66

The analysis phase of the field testing was completed by comparing the "DT" test sequence output reports (using both the HTML and log formats from each tool run) against the earlier benchmarked data reports and bookmarks. The comparative method was found to be the most appropriate way in order to make a determination if each of evidence data groups from Table 3.3 were being accurately reported by the USBForensicReporter© tool.

Table 5.3 presents a summary of the analysis findings after output from the USBForensicReporter© tool was compared against output from the earlier sample toolset benchmarking. Overall, when output data was compared across all five tools the analysis findings determined that the USBForensicReporter© tool was reporting a consistent depth of USB artifacts that was on par with the well-known commercial forensic tools such as the EnCase® Forensic and FTK® toolsets. Each tool in the sample toolset also displayed key evidence related items in slightly different formats and the overall output reporting was inconsistent in nature.

Table 5.3 also identifies that the USBDeviceForensics© tool (Tool 2) does not currently extract any of the drive letter artifacts recorded in the *Windows Portable Devices* subkey even though the SOFTWARE hive from Windows® 7 operating systems is supported in the tool's evidence set selection options. An avenue of key forensic artifact information could be lost if a practitioner is not aware that a record of the last drive letter assignment and associated time stamp values are recoverable from this location for a wide range of USB devices, including USB thumb drives. The field testing further identified that related information is not recorded in this subkey for PSD devices such as the Seagate FreeAgent GoFlex External USB 2.0/3.0 device or the Transcend StoreJet 500 USB 2.0 Portable Storage device used during the earlier toolset evaluations.

Some minor data output discrepancies were also noted for the USBDeview© tool during the analysis comparison. These discrepancies related to the final digit in the *Hardware ID* details in the *USBSTOR* subkey being reported as "1.00" instead of "1.01" and the last written timestamp of the *USBSTOR* subkeys not displaying the second's value when compared to the other USB tools.

Table 5.3

Sample Toolset and USBForensicReporter© Tool Analysis Summary

USB Evidence Group Analysis Details for Comparison	USBDeview Tool (v1.91)	USBDevice-Forensics Tool Tool (v1.0.7)	EnCase Forensic Tool (v6.18)	FTK Forensic Toolkit (v3.3) FTK RegistryViewer (v1.6.3)	USBForensic-Reporter Prototype Tool (v1.0.6)
USB Vendor, Product and Version Details	~	✓	✓	✓	✓
FriendlyName	✓	✓	✓	✓	✓
Device Serial Number	✓	✓	✓	✓	✓
USBSTOR Last Written Timestamp	~	✓	✓	✓	✓
MountedDevices Drive Letter Mapping	✓	✓	✓	✓	✓
Volume GUID Number	x	✓	✓	✓	✓
Device Classes GUID and or Last Written Timestamp	x	✓	✓	✓	✓
NTUSER.DAT – MountPoints2 User Account Mapping	x	✓	✓	✓	✓
Windows Portable Devices Mapping and Last Written Timestamp	x	x	✓	✓	✓
Setupapi.dev.log – 1st Device Connection Timestamp	x	✓	✓	✓	✓

✓ The tool accurately presents the correct data
x The tool does not support reporting of this type of data
~ The tool partially satisfies presenting the correct data

5.4. PRESENTATION OF FINDINGS: USB TOOL DEVELOPMENT TESTING

Section 5.3 analysed and reported on findings from the field testing of the USBForensicReporter© prototype tool. USB devices and evidential datasets collected during the sample toolset benchmarking in Section 4.2 were again used as a baseline of tool performance and functionality testing in the field phase.

Key areas of the performance and functionality testing are reported using a series of tables to provide a visual assessment summary of the developed USB tool against previously collected data from the benchmarked sample toolset.

Table 5.4 presents a summary of the overall processing times captured during field testing of the developed prototype tool. Only seven tests were carried out in the development field testing as the same Seagate FreeAgent GoFlex External USB 2.0/3.0 device was previously used in Tests 6 and 7 during the sample toolset evaluations. Tests 1 to 6 tested USB 2.0 supported thumb drive and PSD devices whilst Test 8 only tested a USB 3.0 supported PSD device.

The findings show that the USBForensicReporter© tool (in its current development state) processed the previously collected suspect registry files and USB device data on an average of *1.25* seconds during field testing. The results also demonstrate to the digital forensic practitioner and researcher alike that automated tools and processes have an advantage over analysis tools that utilise manual reporting processes when multiple USB devices are being examined in one case.

Table 5.4

Comparison of Processing Times Captured During the Development Field Testing

Test Sequence Number	USBDeview Command-Line Reporting	EnCase Forensic Manual Reporting	FTK Registry-Viewer Manual Reporting	USBForensic-Reporter Prototype Reporting
Test 1	01:48.66	11:35.12	03:36.63	00.01.62
Test 2	01:35.62	14:10.80	03:33.61	00:01.10
Test 3	01:24.78	11:05.28	03:28.86	00:01.24
Test 4	01:17.54	10:01.10	03:15.17	00:01.31
Test 5	00:59.87	10:09.60	02:48.10	00:01.45
Test 6	01:04.45	10:32.14	02:43.83	00:01.75
Test 7	01:00.22	10:20.45	02:50.34	Not Conducted
Test 8	No Tool Support for the USB 3.0 Device	10:12.94	02:41.58	00:00.66
Average Time for Tests	01:18.73	11:00.93	03:07.27	00:01.25

Processing Times for the Benchmarked Sample Toolset

Processing Times for the Development Field Testing

Tables 5.5 and 5.6 present two comparison matrixes of the earlier toolset examples against the developed USBForensicReporter© tool. The evaluation criteria that were used in the earlier toolset benchmarking of Chapter 4 were again used during the development field tests to measure overall performance and functionality of the USBForensicReporter© tool under standardised test conditions.

Figure 5.5 presents results relating to all of the tools being tested on USB 2.0 supported devices, whilst Figure 5.6 presents results relating to the tools being tested on the single Seagate GoFlex USB 3.0 supported PSD device. When combined, the conditional requirements results demonstrate that the USBForensicReporter© tool matches the performance and functionality of well-established and widely used forensic software toolsets, namely EnCase® Forensic and FTK®/FTK® RegistryViewer®.

Table: 5.5

Prototype Field Testing Comparison Matrix Conducted for USB 2.0 Supported Devices

Conditional Requirements (Testing Criteria)	**Evaluated Tools**				
	USBDeview	**USBDevice-Forensics**	**EnCase Forensic**	**FTK/FTK Registry-Viewer**	**USB-Forensic-Reporter**
CR1 - The tool supports processing of digital source evidence (i.e. evidence file format or individual Live and Offline Registry hive data)	✓	✓	✓	✓	✓
CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard	✓	✓	✓	✓	✓
CR3 - The tool supports the examination and reporting of USB 2.0 devices	✓	✓	✓	✓	✓
CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool	~	~	✓	✓	✓
CR5 - The original digital source evidence is unchanged by any subsequent tool activity or user actions	✓	✓	✓	✓	✓
CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user	**N/A**	**N/A**	✘	**N/A**	**N/A**
CR7 - If the tool logs processing information, the information is accurately recorded in a log file and or screen output for the user	✓	✓	✓	✓	✓
CR8 - The tool allows extraction of analysis and log information into a format that is viewable by the user	✓	✓	✓	✓	✓

✓ The tool fully satisfies the conditional requirements testing criteria

✘ The tool does not satisfy the conditional requirements testing criteria

~ The tool partially satisfies the conditional requirements criteria

N/A The tool does not support the conditional requirements or it was not encountered

Table: 5.6

Prototype Field Testing Comparison Matrix Conducted for a USB 3.0 Supported PSD Device

Conditional Requirements (Testing Criteria)	**Evaluated Tools**				
	USBDeview	**USBDevice-Forensics**	**EnCase Forensic**	**FTK/FTK Registry-Viewer**	**USB-Forensic-Reporter**
CR1 - The tool supports processing of digital source evidence (i.e. evidence file format or individual Live and Offline Registry hive data)	✘	✓	✓	✓	✓
CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard	✘	✓	✓	✓	✓
CR3 - The tool supports the examination and reporting of USB 3.0 devices	✘	✓	✓	✓	✓
CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool	✘	✓	✓	✓	✓
CR5 - The original digital source evidence is unchanged by any subsequent tool activity or user actions	✘	✓	✓	✓	✓
CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user	✘	**N/A**	**N/A**	**N/A**	**N/A**
CR7 - If the tool logs processing information, the information is accurately recorded in a log file and or screen output for the user	✘	✓	✓	✓	✓
CR8 - The tool allows extraction of analysis and log information into a format that is viewable by the user	✘	✓	✓	✓	✓

✓ The tool fully satisfies the conditional requirements testing criteria

✘ The tool does not satisfy the conditional requirements testing criteria

~ The tool partially satisfies the conditional requirements criteria

N/A The tool does not support the conditional requirements or it was not encountered

5.5. CONCLUSION

Chapter 5 reported the development, validation and evaluation testing of the USBForensicReporter© prototype tool. Variations to the software design and data requirements proposed in Chapter 3 were summarised and then implemented in testing methods used throughout the field and analysis phase of the tool development. General tool operation and data reporting was also discussed in order to showcase a number of unique USB analysis and reporting features that were not found in any of the other evaluated USB analysis tools from the toolset evaluations in Chapter 4.

The unique features utilise the implementation of an innovative comparative analysis function in the coding and data output formats of the tool. The software function uses a combination of suspect Registry files and the Registry files from the analyst's workstation computer in order to make an accurate determination if a physical USB storage device had previously been connected to the suspect computer system. Data output is captured and reported in an HTML report format with the novel use of a series of colour-coded panels and *Device Alert* fields to detail USB storage device information and *positive* or *negative* connectivity between the physical suspect USB storage device and suspect Registry files.

Collected forensic data derived from the benchmarked sample toolset evaluations was utilised in field testing of the USBForensicReporter© tool to determine overall performance and functionality of the developed USB tool. The field findings showed that the prototype tool is capable of processing USB information and related forensic artifacts in a rapid manner whilst on an equal footing with other benchmarked and industry accepted USB analysis and forensic reporting tools. The findings in Chapters 4 and 5 will be now used in Chapter 6 as a foundation for further discussion on the research outcomes and to test the main hypotheses statement and subsequently answer the research questions previously formulated in Chapter 3.

Chapter 6

Discussion

6.0 INTRODUCTION

Chapters 4 and 5 conveyed significant research findings relating to the benchmarking of a sample USB toolset and the development of a new USB forensic analysis and reporting tool in order to provide the basis for answering the research questions and overall hypothesis. Each of the completed evaluation, software design and tool development phases was essential to gaining an in-depth understanding of the area of research whilst also providing a mechanism for the creation of the USBForensicReporter© tool.

Chapter 6 provides a connection between the previous two chapters whilst discussing the results from each of the main research phases for final research conclusions to be drawn from. Evidence collected during the literature review, USB toolset evaluations and development phases will assist in responding to the main research question and related sub-questions and to ultimately test the hypothesis statement.

Section 6.1 discusses key findings from the USB toolset evaluations, gap analysis and tool development phases to make a determination on the overall success of the research project and the impact that a new analysis and reporting tool has on current tool offerings, and for future USB based memory device examinations. Limitations in the current research and tool design are summarised in Section 6.2. Section 6.3 concludes the chapter by discussing answers to the research questions and hypothesis developed in Section 3.2.

6.1. DISCUSSION OF RESEARCH FINDINGS

Chapters 4 and 5 reported on different aspects of the research findings relating to the sample toolset evaluations and development of a new forensic analysis and reporting prototype tool. Section 6.1 will now discuss three of the core areas that connected the different findings from each chapter together.

The three core areas are: the sample USB toolset evaluations; the dual gap analysis phases, each of which was critical to evaluating tool performance and identifying gaps in functionality and usability, and the software development associated to the new tool. Important results from each area will be discussed in order to lay the foundation for answering the research questions and proving or refuting the hypothesis statement in Section 6.3.

6.1.1. Sample USB Toolset Evaluations

Section 2.4 previously identified that no past studies or tool testing regimes could be found in relation to the benchmarking of forensic or IT-based USB data analysis and reporting tools. The tool evaluation findings in Section 4.2 were therefore critical in underpinning the foundations of the current research. The findings also provided a standard benchmarking mechanism for the design, validation, verification and testing of the USBForensicReporter© tool development life cycle. The USB tools used in the sample toolset evaluations provided an ideal mix of commercial and free-ware tools for testing purposes. Each of the tools is commonly utilised in differing capacities across a variety of law enforcement and corporate digital forensic laboratories in New Zealand.

A set of eight conditional requirements were developed in Section 3.3.3 to form the basis of the tool evaluation method. The conditional requirements established a set of assertions or conditions to measure tool functionality, analytical process and output reporting for each of the sample tools. These requirements were derived in part from a set of industry-recognised testing criteria (specifically NIST assertions DA-AM-01 to DA-AO-24) that had earlier been developed by NIST (2005). Whilst not adhering strictly to the NIST based digital data acquisition tool testing assertions, implementation of the proposed evaluation criteria proved to be successful in determining if the USB tools in the sample toolset were able to conform to each of the conditional requirements.

The analysis of results from the sample toolset evaluations presented in Tables 4.3 and 4.4 provided an insight into the overall performance capabilities of four sample tools that are frequently found in digital forensic laboratory environments. Data was collected, analysed and reported from each of the tools under evaluation.

The USBDeview© tool (designated as Tool 1) consistently performed at the lower end of the evaluation results as the data collection functionality of the tool failed to meet the CR4 criteria. This may be explained by the USBDeview© tool only being designed to examine the *SYSTEM* registry file, and also not supporting the reporting of newer USB 3.0 storage devices at the time the toolset evaluations were undertaken. The limited reporting capability of the tool meant significant USB data attribute information was subsequently overlooked. These areas include the *Setupapi.dev.log* that records the 1st time a USB storage device is connected to the system; linkage of past USB device connection to a user account via entries in the *MountPoints2* subkey of the user's *NTUSER.DAT* file, and confirmation of recent drive letter assignments from entries in the *Windows Portable Devices* subkey of the *SOFTWARE* hive file.

From the perspective of forensic use, the USBDeview© tool (in its evaluated form of version 1.91) would still be capable of doing the job under field conditions where no commercial USB protocol analyser (such as the USBlyzer software) or forensic software was available. The tool's strengths are that it can quickly report on USB and "live" *SYSTEM* related information from the connection of a physical USB storage device (whilst utilising appropriate write-blocking devices and precautions) or remote storage location. Overall, the USBDeview© tool was found to be best suited for general IT system administration tasks. These tasks could include the identification of connected USB devices on "Live" computer systems, and auditing of extracted *SYSTEM* hive files via the command-line analysis option.

The benchmark findings of the USBDeviceForensics© tool (designated Tool 2) identified that the tool accurately reported a greater range of USB forensic artifacts across all of the conditional requirements when compared to the USBDeview© tool. The only exception was that CR 4 was only partially met due to non-reporting of drive letter artifacts contained within the *Windows Portable Devices* subkey of the S*OFTWARE* hive file. The tool's interface did however allow for the SOFTWARE hive file to be selected along with the other *SYSTEM*, *NTUSER.DAT* and *Setupapi.dev.log* files.

Limitations to the forensic collection capability were particularly noticeable in Tests 1 to 4 of the sample toolset evaluations when USB 2.0 supported thumb drive devices were being examined by the tool. Manual examinations of each of the *Windows Portable Devices* subkeys in the *SOFTWARE* hive files identified the presence of recent drive letter assignments and device information for each of the USB 2.0 thumb drives used in these test sequences. Not being able to report on USB artifacts from this subkey was a missed opportunity for obtaining further evidential source data relating to recorded recent drive letter allocations.

Law enforcement and corporate investigators frequently request digital forensic practitioners to identify drive letter assignments relating to specific USB devices under examination or data review. The inability to identify a recent drive letter for a particular USB thumb drive device could have an impact on examination outcomes. The impact is particularly relevant when drive letters are reassigned in the *MountedDevices* subkey of the *SYSTEM* hive file by more recent USB connections or if past assignment records are overlooked in the *Windows Portable Devices* subkey of the *SOFTWARE* hive file by the practitioner.

According to Lee (2009) and Carvey (2009) the *Windows Portable Devices* subkey is a relatively new USB evidence source for practitioners examining Windows Vista® and Windows® 7 operating systems as it was non-existent in the Windows® XP operating system. Device references and drive letter assignments for all of USB thumb drives used in the toolset evaluations were recorded in this subkey location. Alongside the drive letter and USB thumb drive information, Carvey also identified that this subkey location can contain information for other USB device types such as iPods and digital cameras.

USBDeviceForensics© tool results from Tests 5 to 8 and presented in Table 4.3 were not affected by the identified *SOFTWARE* analysis limitation. The toolset evaluations determined that recent drive letter assignments for PSD devices containing conventional hard drives (i.e. 2.5 inch or 3.5 inch standardised form factors) are not recorded in the *Windows Portable Devices* subkey by Windows® 7 operating systems. Drive letter assignments are only found in the *MountedDevices* subkey of the *SYSTEM* hive file. Both the *MountedDevices* and *Windows Portable Devices* subkeys therefore play a pivotal role in allowing past drive letter assignments and device information to be associated to physical USB storage devices.

The *MountedDevices* subkey information can also be equally challenging to the digital forensics practitioner as related value names and binary data can be displayed in different formats. To illustrate the point, Figure 6.1 displays the name and data value formats from the *MountedDevices* subkey for a Seagate FreeAgent GoFlex PSD USB 2.0/3.0 device and SanDisk Cruzer USB 2.0 thumb drive used during the toolset evaluations.

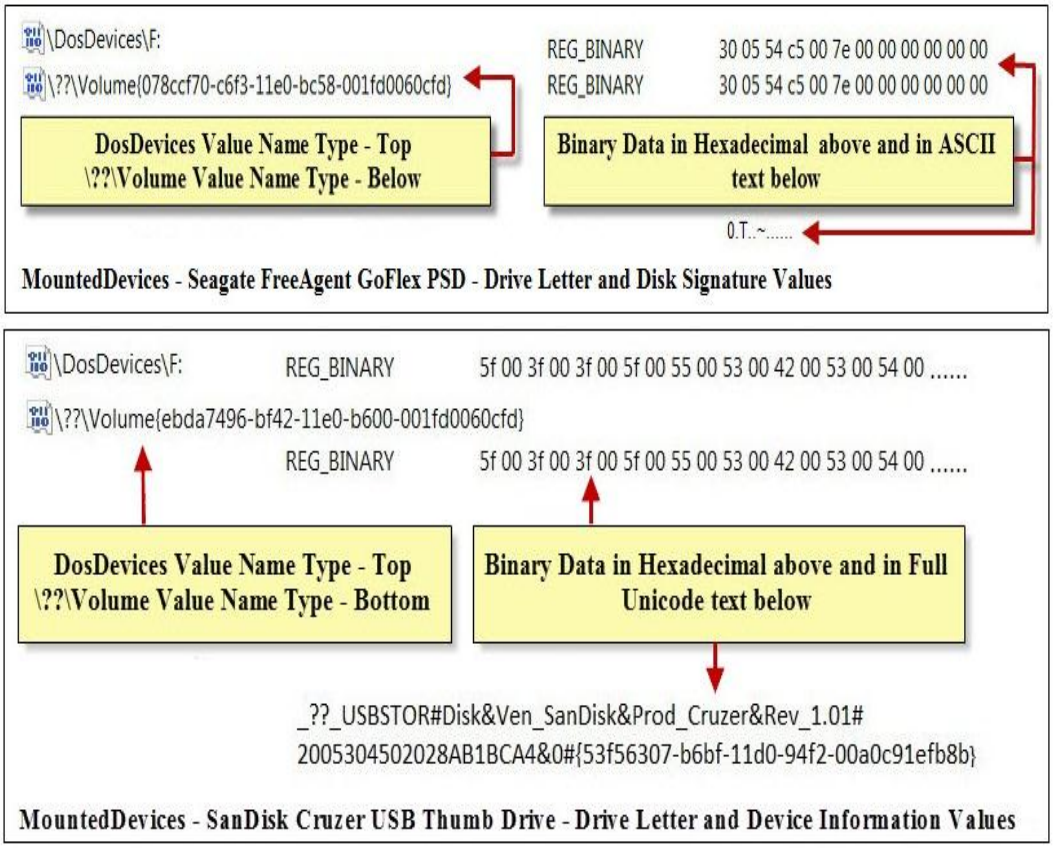


Figure 6.1. MountedDevices Subkey Variations in USB Device Types

In both examples the Windows® operating system assigned drive letter is displayed under the *\DosDevices\Letter#:* name value. PSD devices containing conventional hard drives utilise a corresponding binary data value containing the disk signature which is 12 bytes in length (refer to the Seagate FreeAgent GoFlex REG_BINARY details in the top portion of Figure 6.1). Non-PSD devices such as USB thumb drives do not contain a disk signature but consist of larger binary data values containing the relevant device class and unique instance identifiers associated to the USB device (refer to the SanDisk Cruzer REG_BINARY details in the lower portion).

The disk signature of PSD devices and Windows-based hard drives containing the NTFS file system can be determined at decimal offset 440 within the Master Boot Record (MBR) as shown in Figure 6.2.

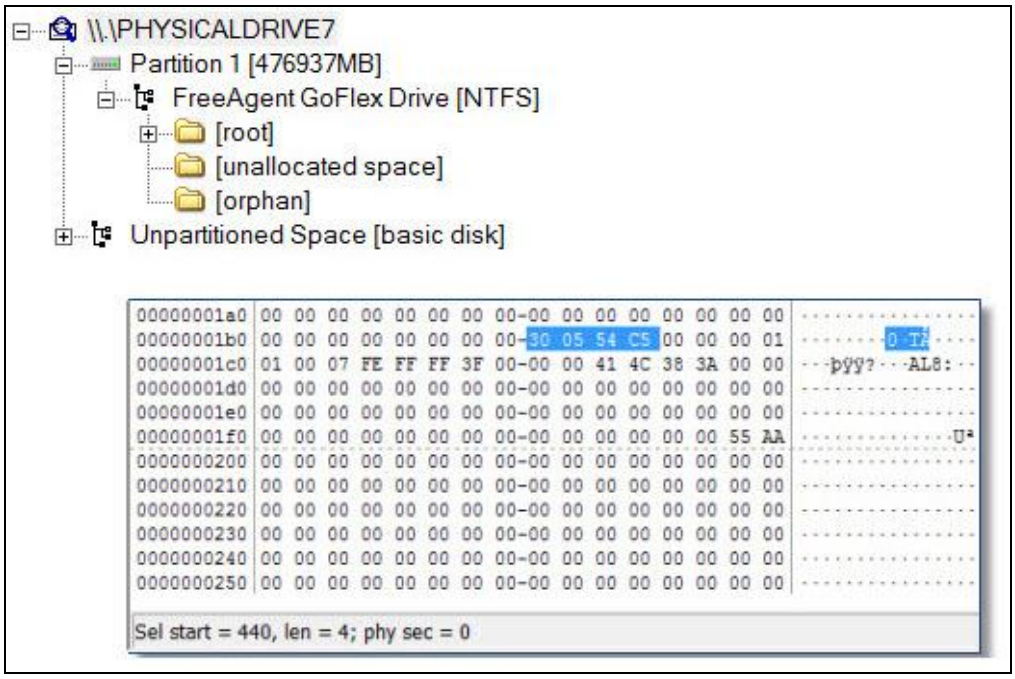


Figure 6.2. MBR Disk Signature Correlation Using FTK® Imager Forensic Software

Disk signatures for PSD and hard drives can also be found in *Windows Link File* data relating to documents and other data files in order to associate a particular USB storage device with user activity if drive letter assignments have been reused by other devices in the *MountedDevices* subkey.

Notwithstanding the lack of full *SOFTWARE* file support, the overall findings for the USBDeviceForensics© tool evaluations identified that the tool provided comprehensive output reports containing the vast majority of the USB forensic artifacts identified in Table 3.2 for USB thumb drives. Reporting did however come with some added labour costs or overheads to the tool user. The labour costs were associated to additional processing action being required on the user's part to modify both the text and CSV output reports in a manner that made each of the output formats more presentable and easier to understand. Whilst not a major factor in the current toolset evaluations where only a small number of USB devices were being examined, it would be a different matter if multiple or remote computer systems of interest to an investigation generated additional processing overheads.

Extra information retrieval and reporting overheads would make the more labour-intensive tool a less effective option in a corporate environment where investigative costs usually have to be kept to a minimum (Pangalos, Ilioudis & Pagkalos, 2010).

The overall findings for the EnCase® Forensic (Tool 3) and FTK®/FTK® RegistryViewer® (Tool 4) tool evaluations showed that commercially produced forensic tools do have an advantage over smaller niche markets or freeware IT and forensic products such as the USBDeview© and USBDeviceForensics© tools. With the exception of the EnCase® tool facing some performance issues in Test 2 of the toolset evaluations, both tools provided the most comprehensive benchmarking scores and evidential groups of USB artifacts for a digital forensics practitioner to analyse and report on.

The broad range of design functionality, processing and reporting options (both from analytical and forensic imaging view points) also give established forensic tools such as EnCase® and FTK® an advantage over other lesser known and specialised competitors by making them very popular and widely accepted across the international digital forensics community. Likewise, this can also be a disadvantage to some smaller industry professionals as the cost of most commercial tools can generally make them all-but prohibitive in comparison to larger and more well-funded law enforcement and corporate laboratories.

In summary, the evaluation criteria and tool evaluation templates were both successfully implemented and reported in the sample toolset evaluation phase of Section 4.2.2. The findings from the evaluations assisted in answering a number of the research sub-questions. Registry data forming the reference list in Table 3.2 supported the identification and collection of past USB activity on the test computer systems. The references assisted with answering research sub-questions 2 and 3 in Section 6.3.2 by pinpointing important Windows® Registry and Windows® 7 specific artifacts and could be used as an aid by other practitioners in future USB examinations. Research sub-question 4 in Section 6.3.2 was answered by conducting the sample toolset evaluations in-order to create a published benchmark of commonly used USB analysis and reporting tools in New Zealand.

The USB evaluation criteria and evaluation templates now provide digital forensic practitioners with a new tool testing methodology to employ alongside other established digital forensic testing criteria and methods for both software and hardware applications.

6.1.2. Gap Analysis

The gap analysis findings were analysed in Section 4.3 and presented in Tables 4.4, 4.5, 4.6, and Figures 4.7 and 4.12. The gap analysis methodology provided an ideal assessment instrument for creating a benchmark of actual tool performance in the sample toolset evaluations. Phase One was important for collecting data to produce the gap analysis matrices presented in Tables 4.4 and 4.5. Both matrices presented a summary of the sample toolset evaluation results in a readable and easily understandable format. Each individual test and tool were matched against the eight conditional requirements through the use of symbols (i.e. tick, cross, tilde and N/A). In essence, these symbols represent an overall pass, fail, partial or not applicable fulfilment for each of the evaluation requirements so as to create a benchmark standard of performance for the evaluated toolset.

The Phase One Gap Analysis ultimately allowed the USB toolset evaluation gap identified in literature review of Chapter 2 to be filled. Evaluating both the strengths and weaknesses associated with each tool under controlled laboratory conditions further assisted in answering research sub-question 4. The Phase Two Gap Analysis provided tool specific findings presented in and Table 4.6, and Figures 4.7 and 4.12. This phase was critical to the research as it provided a mechanism for identifying gaps or weaknesses in the existing toolset so improved features could be incorporated into the design phase of the USBForensicReporter© tool. A unique categorisation and scoring method provided an ideal assessment instrument so each of the sample USB tools could be more precisely gauged against the six Gap Analysis categories developed in Section 4.3.2. The findings from Section 4.3.3 identified three significant gaps relating to completeness of evidence group reporting, the provision for multiple reporting formats, and the simplification of the data output and additional actions by the tool user.

The first gap identified that the USBDeview© and USBDeviceForensics© tools were unable to report all of the evidence groups presented in Table 3.2.

The second gap identified that the FTK® RegistryViewer® tool provided only one HTML reporting format that was unable to be edited for further reporting purposes. Therefore, the prevision for multiple reporting formats and depth of reporting offered by the EnCase®, USBDeviceForensics© and the FTK® tools was considered the optimum level for future USB tool design and functionality development.

The final gap identified that the USBDeviceForensics© and EnCase® Forensic reporting formats required additional formatting and reporting time in each of the tools being evaluated in order to obtain a fully functional output report. Each of the identified gaps pointed to the need for several improvements and new features to be implemented in the respective analysis and reporting features of the USBForensicReporter© tool design, therefore answering research sub-question 5 in Section 6.3.1 and the main research question in Section 6.3.2. The findings in Section 4.3.3 also demonstrated that the chosen gap analysis method could be used by forensic software developers to identify significant weaknesses and limitations in existing toolsets.

6.1.3. Developed USB Prototype Tool

Field testing and analysis findings produced as part of the USBForensicReporter© tool development process were presented in Sections 5.3 and 5.4. The tool design proved to be very stable in operation and produced reliable output results during final prototype testing, analysis and verification phases of the chosen SDLC model. Discussion will focus here on key aspects of the developed prototype tool in order to provide a foundation for answering research sub-questions 5, 6 and 7, and the main research question.

The ability of the USBForensicReporter© tool to conduct a USB storage device based comparison analysis against a suspect evidence set of Windows® 7-based Registry files was found to be one of the most important aspects of the tool's functionality. None of the evaluated tools allowed for the physical analysis and reporting of a suspect USB device against the evidence set whilst utilising one tool interface. The comparative analysis method is therefore considered to be an innovative design feature which on its own merits is an improvement over the analytical functionality of existing USB forensic tools.

The improvement of tool functionality and USB forensic examination methods as a result of the tool development process assisted in addressing research sub-questions 5 and 7, and the main research question of the current research.

The implementation of a clean looking and editable HTML report containing USB device information and the reported *Device Alert Status* comparative-analysis results is another original and important aspect of the reporting features offered by the USBForensicReporter© tool. The individual *Device Alert Status* fields and section panels for each reported USB device from the suspect *USBSTOR* registry hive location are colour-coded to emphasize a level of high (red) or low (blue) importance for that particular device. The coloured panels quickly identify if a suspect USB device has or has not previously been connected to the suspect computer system under examination.

USB devices with red highlighted section panels and corresponding *Device Alert Status* fields containing the **Alert Device: Found** entry provide a visual identification that such a USB device **matched** Windows registry entries found in the suspect evidence set. The comparative analysis match signifies to the practitioner that the USB device is important and may require further analysis and review to determine the evidential value of the data content. The feature would be particularly beneficial when examining multiple USB devices in criminal investigations where the presence of child pornography is suspected or where data theft is alleged in corporate investigations.

USB device details containing blue highlighted section panels and *Device Alert Status* fields with a corresponding **No Matches** entry quickly identify that the physical suspect USB device has **not** previously been connected to the computer system under investigation. None of the sample toolset evaluations presented in Chapter 4 contained this type of analytical and unique reporting functionality.

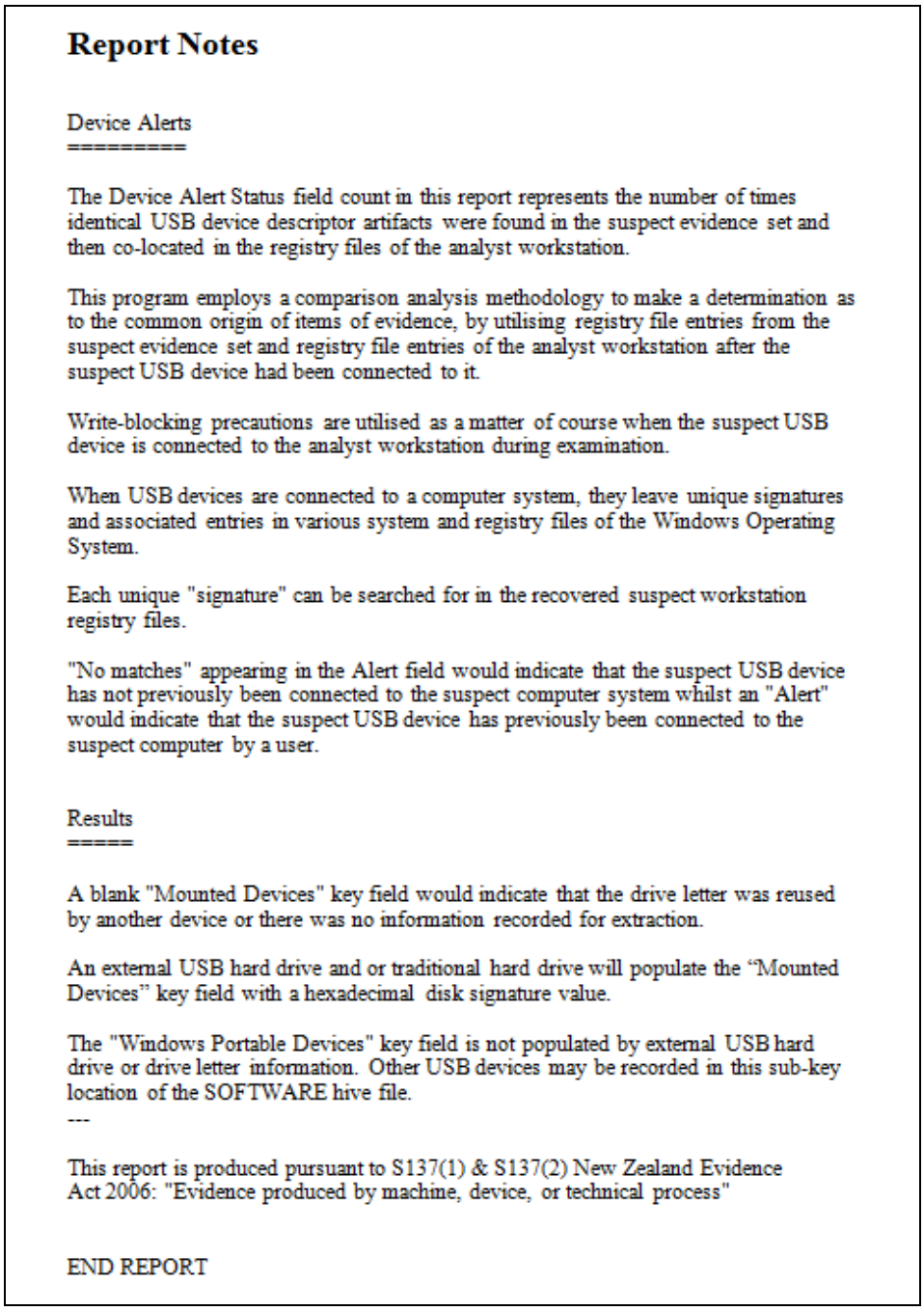
A further feature of the tool's HTML reporting format was inclusion of a *Notes Section* at the end of the USB Device Details section. Commercial forensic tools such as FTK® RegistryViewer® offer a professionally branded HTML report format with the provision for large amounts of USB artifacts data to be selected and printed; however, no explanation is provided about key elements of the report for the reader to digest.

More often than not these types of standalone HTML reports require one or more bookmarked sections to be copied and pasted into conventional Microsoft® Office documents so explanations can be added by the digital forensics practitioner. The explanations allow the practitioner to highlight USB concepts, various Registry file locations and processes relating to the discovery of USB artifacts that may be of evidential value to investigators.

The inclusion of a legal statement in the notes section (in this case the New Zealand Evidence Act 2006 wording for "machine produced evidence") is also a localised example of how legal requirements or reporting standards can be incorporated into the HTML output. With the additional insertion of a sworn statement and signature block the current HTML report could be easily filed as part of a formal written statement in criminal proceedings or as part of a signed affidavit in civil proceedings here in New Zealand.

Figure 6.3 provides an extract from the current *Notes Section* of the USBForensicReporter© tool. The section is comprehensive and free flowing, yet easily adaptable by future tool users to better reflect organisational standards of reporting and to the level of computer literacy and knowledge that investigators reviewing the output could potentially have. Enhancement features such as the notes section and the implementation of innovative *Device Alert* Status comparative analysis method in the design and functionality of the USBForensicReporter© tool further assisted in providing answers for the main research question and sub-questions 5 and 7.

Finally, data protection features in the design of the prototype tool helped in answering research sub-question 6 in Section 6.3.1. Data integrity is maintained during tool operation via a number of built-in protection mechanisms. The first mechanism utilises the Windows file attribute by automatically changing each of the extracted *SYSTEM*, *SOFTWARE*, *NTUSER.DAT* and *setupapi.dev.log* evidence files to read-only prior to further file mounting and processing action. An industry-standard mathematical algorithm known as Message Digest 5 (MD5) was also utilised in the design of the backend tool processing module to maintain the validity of the evidence file set throughout each stage of data processing and reporting.



Report Notes

Device Alerts
=========

The Device Alert Status field count in this report represents the number of times identical USB device descriptor artifacts were found in the suspect evidence set and then co-located in the registry files of the analyst workstation.

This program employs a comparison analysis methodology to make a determination as to the common origin of items of evidence, by utilising registry file entries from the suspect evidence set and registry file entries of the analyst workstation after the suspect USB device had been connected to it.

Write-blocking precautions are utilised as a matter of course when the suspect USB device is connected to the analyst workstation during examination.

When USB devices are connected to a computer system, they leave unique signatures and associated entries in various system and registry files of the Windows Operating System.

Each unique "signature" can be searched for in the recovered suspect workstation registry files.

"No matches" appearing in the Alert field would indicate that the suspect USB device has not previously been connected to the suspect computer system whilst an "Alert" would indicate that the suspect USB device has previously been connected to the suspect computer by a user.

Results
=====

A blank "Mounted Devices" key field would indicate that the drive letter was reused by another device or there was no information recorded for extraction.

An external USB hard drive and or traditional hard drive will populate the “Mounted Devices” key field with a hexadecimal disk signature value.

The "Windows Portable Devices" key field is not populated by external USB hard drive or drive letter information. Other USB devices may be recorded in this sub-key location of the SOFTWARE hive file.

This report is produced pursuant to S137(1) & S137(2) New Zealand Evidence Act 2006: "Evidence produced by machine, device, or technical process"

END REPORT

Figure 6.3. HTML Report Extract – Notes Section

To further ensure data integrity and accuracy of reported output, the validation component of the VV method was implemented in the chosen software life cycle so results could be reported in Section 5.2.2. The validation method assessed the tool at a raw Windows registry level to ensure the correct data was being accurately identified and parsed by the tool’s programing code. After the tool was successfully validated, the verification component of the VV method was also tested and reported in Section 5.2.3.

USB artifacts were also successfully verified by utilising another benchmarked forensic tool (in this case the EnCase® Forensic tool) to confirm the accuracy and completeness of the tool's reported output.

6.2. RESEARCH LIMITATIONS

Limitations to the proposed research methodology were previously identified in Section 3.5. The limitations related to the initial toolset evaluations and development of the USBForensicReporter© tool. The research methodology evolved over time from a largely theory-driven concept to a more experimental and design-focused study as the toolset evaluations and SDLC phases were implemented. Differing impacts on these limitations were experienced along the way and additional limitations were also discovered by the researcher that was not originally envisioned. Each limitation and impact on the benchmark sample toolset findings presented in Chapter 4 and the developed prototype tool discussed in Chapter 5 will now be examined.

6.2.1. USB Tool Evaluations

The first limitation related to the selection of the sample tools for the toolset evaluations and benchmarking processes. The evaluations were limited to general USB analysis and reporting features only as the four tools originally chosen for testing purposes were very diverse in nature and functionality. The USBDeview© and USBDeviceForensics© tools were freeware and USB feature specific by design. Both tools had not previously been used by the researcher and were therefore blind-tested throughout the toolset evaluations. The EnCase® Forensic and FTK® tools were more fully-featured commercial forensic software platforms with an array of imaging, analytical and reporting options being made available to the user.

The results in Section 4.2 demonstrated that although the USBDeview© tool was able to report basic USB information for a connected device or remote offline evidence set it was very limited in its forensic capability as a comprehensive analysis and reporting tool for USB based forensic examinations. The narrow focus of the tool's analytical capacity (i.e. being limited to *SYSTEM* registry file and reporting of USB 2.0 supported devices only) was unforeseen at the time of selection.

The tool's lack of suitability as a broader forensic analysis and reporting instrument therefore impacted on the potential for more comprehensive data sets to be collected during each of the toolset evaluations. In hindsight, such a limitation may have been reduced if a more expansive range of industry practitioners had been surveyed on both primary and secondary tool usage to identify a greater selection of tools to choose from.

Likewise, more preliminary tool testing could have been conducted by the researcher to determine a basic level of common tool operability during the initial tool selection stage. The preliminary testing may have eliminated tools that were not suitable for comprehensive digital forensic related examinations. Selecting such a level of common operability and functionality may have assisted in providing a more consistent range of reported data results from the Test 1 sample toolset evaluations.

6.2.2. USB Storage Device Limitations

The second limitation relating to the tool evaluations was the use of both USB 2.0 and USB 3.0 storage devices. The data collection phase of the research utilised a combination of four USB 2.0 thumb drive devices and two USB 2.0 external PSD devices, one of which (the Seagate FreeAgent GoFlex device) operated under both USB 2.0 and USB 3.0 conditions. In a perfect world, an optimum USB test environment for performing stabilised scientific-based experiments would ideally consist of test apparatus comprising entirely of USB 2.0 storage devices or USB 3.0 storage devices to maintain standardised testing requirements.

From a New Zealand perspective, USB 3.0 thumb drive support was simply not cost effective enough for inclusion in the test environments at the time the research methodology was formalised in February 2011. Anecdotally in December 2011 availability of USB 3.0 thumb drives was still somewhat limited and expensive when compared to USB 3.0 external PSD devices and the more readily available USB 2.0 device technologies.

The researcher also had some concerns in the earlier stages of research design about the use of mixed apparatus in the collection and testing phases. The concerns were primarily around the potential need for future researchers to require different hardware resources in order to evaluate the test results or replicate the findings when USB 2.0 devices are superseded by the newer USB 3.0 technology.

Hardware requirements and test environments can also be difficult to replicate within or across regional and organisational boundaries as funding for scientific research and hardware procurement may be difficult to obtain from existing budgets in both private and public forensic laboratories.

Similar factors were identified in earlier tool testing and forensic research by established measurement and standards organisations such as NIST. NIST (2002) found that there needs to be some flexibility when replicating test environments as "available hardware determines the strategy for organizing the test process" (p.37). Hardware used in the original test methodology may have been superseded by newer technologies or is simply not available to the individual tester or testing body so "substitutions must be made to run the test cases" (NIST, 2002, p.38). The test results in the current research were only affected by use of the mixed-apparatus method in Test 8 of the sample toolset evaluations.

The USBDeview© tool evaluations established that the tool design was limited to supporting USB 2.0 storage devices only. No test results could be obtained for that particular tool in Test 8 whilst utilising the USB 3.0 plug–and–play connectivity features of the Seagate GoFlex PSD test device. Overall, the sample toolset evaluations and selected USB devices worked well together in the current research, providing a robust method that could be applied to future USB research when USB 3.0 storage devices are more readily available and utilised by computer users.

6.2.3. Software Design Limitations

The software design phase of the prototype tool development also identified two new limitations that were unforeseen when the software design life cycle was proposed in Section 3.3.3. The first design limitation was the mounting of the widely accepted EnCase® evidence file format or Expert Witness Format (EWF) to allow direct processing by the USBForensicReporter© tool. The evidence mounting option was unable to be completed during the tool development testing due to technical and coding issues being encountered during the loading of the evidence container on a test computer system running Windows® 7. The complied test code simply could not load the evidence container under preliminary test conditions.

In order to allow the suspect registry set to be processed by the USBForensicReporter© tool other more reliable and proven techniques were investigated to find a temporary solution for the problem.

The mounting of suspect registry files was subsequently redesigned to utilise the established *Load Hive* feature of the Windows® Regedit tool. The *Load Hive* feature allows the selected suspect registry files to be loaded as temporary and standalone keys within the Windows® 7 Registry of a forensic workstation for further tool processing. Further research is required to find more permanent solutions for EnCase® evidence support and loading of the suspect evidence files that can more easily be integrated into the processing code of the USBForensicReporter© tool.

The second limitation of the software design was the sole use of the serial number (taken from the device's *iSerialNumber* and commonly referred to as the *UIID* by Carvey (2011*)* as the main search string for both processing and comparison analysis purposes in the USBForensicReporter© tool. Section 2.1.2 identified that the unique serial number of a USB device can greatly assist digital forensic practitioners in linking physical USB devices to a particular Windows® 7 computer system via an array of registry artifacts and log file entries. Furthermore, Section 2.2.2 also identified that if the manufacturer of the USB device did not include a serial number in the embedded device descriptor information of the device then the Windows® Plug-and-Play Manager would generate a *ParentIdPrefix* value in its place. The developed prototype tool currently reports all devices found in the *USBSTOR* subkey including USB devices identified with *ParentIdPrefix* values but is not specially designed to search the other registry files for *ParentIdPrefix* values.

Results from the field testing determined that all of the randomly selected USB storage devices used during the toolset evaluations contained unique *iSerialNumber* values only. *Device descriptor* information captured by the USBlyzer© software protocol analyser and reported output from the USBForensicReporter© prototype (Refer to Appendix C for printouts) also reflected that the *iSerialNumber* information was accurately reported.

Such reliable outcomes justified the choice of a single search string source for the current tool design and allowed the development testing to be completed in a timely manner. Provision for full *ParentIdPrefix* processing still needs to be included in on-going and future development of tool so older USB devices that do not contain an embedded serial number are not overlooked as a potential evidence source.

6.3. RESEARCH QUESTIONS AND HYPOTHESIS

The main research question and seven sub-questions were derived from the literature review presented in Chapter 2. The research sub-questions provided a systematic approach for answering the main research question in order to ultimately prove or refute the hypothesis statement. The questions and hypothesis are answered in the following three subsections.

6.3.1. The Research Sub-Questions

Seven individual sub-questions were defined in Section 3.2.2 to provide answers that would collectively lead to the main research question and hypothesis being answered. These sub-questions allowed the researcher to form a better understanding of the different USB aspects that make up the wider research objective. Each sub-question will be answered separately with supporting evidence provided from Chapter 2 and Section 6.1.

Sub-question 1: What is the current state of forensic research related to USB storage devices?

The current state of forensic research involving USB storage devices was determined by the literature review presented in Chapter 2. Overall, the majority of past USB based forensic research was found to be related to the now outdated Windows® XP operating system. Windows® 7 based-research is still in its infancy but growing. Researchers such as Carvey (2011) have also determined that core components of the Windows® Registry are stable and have not changed significantly as newer operating system versions have been released.

Consistency in the manner Windows® Registry files and associated artifacts are stored has also meant researchers, practitioners and developers have a level baseline of knowledge to work from when performing experimental research, case examinations and tool development. Likewise, Non-Windows USB research involving Linux® and Macintosh® based operating systems was also found to be on the rise with comparable USB examination and artifact recovery principals being able to be applied across cross-operating system platforms.

In summary, several problem areas were identified in Section 2.4 that had the potential for further research possibilities. The areas include the lack of depth in Windows® 7 research, the lack of published toolset evaluation results, and the lack of frameworks for USB examinations. The current research has explored and tested the three areas by designing a USB framework, conducting formal toolset evaluations, and developing a prototype tool that would enhance USB based memory device investigations.

Sub-question 2: What operating system records are generated by USB activity on a Windows® computer system?

A theoretical understanding of how the various Windows® operating system versions report the connection of USB storage devices was gained from past research by Carvey and Altheide (2005), Farmer (2007), Lee (2009) and Carvey (2009; 2011). This resulted in the construction of a common list of Windows registry locations and system file records (Table 3.2) to assist other researchers and digital forensic practitioners in obtaining a reference list of USB artifacts. Further field testing also enabled a more in-depth action-based examination of the Windows registry to validate the common keys, subkeys and values associated to USB related artifacts. The testing also identified that *device descriptor* serial numbers for USB storage devices were recorded in the *SOFTWARE\Microsoft\ Windows NT\CurrentVersion\EMDMgmt* subkey. The subkey location has been in existence since the release of Windows Vista® but has not been widely published by digital forensic researchers.

Sub-question 3: What specific Windows® 7 Registry evidential related artifacts can assist a forensic practitioner in USB Examinations?

The fundamental registry artifacts provided in Table 3.2 were expanded during the literature review and methodology design to encompass a more Windows® 7 specific guide to USB forensic artifacts. Specific evidential and forensic data types encompassing the various *SYSTEM*, *SOFTWARE*, *NTUSER.DAT* registry components and *setupapi.dev.log* system file locations of a Windows® 7 based operating system were provided in Table 3.2 to answer the research sub-question.

The inclusion of CR4 (i.e. All common Windows® 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool) in the evaluation criteria for the toolset and tool development evaluations (Tables 4.4 and 4.5) also assisted in answering research sub-question 3. Each of the data sets and output reports from the toolset evaluations, field testing and tool development phases were also verified to ensure that all of the evidence data types and locations were being accurately reported by the each of the evaluated tools and the USBForensicReporter© tool.

Sub-question 4: What forensic or commercial tools examples are currently available to the examiner for collecting and reporting on USB artifacts?

Research sub-question 4 can be answered by the research's utilisation of a formal toolset evaluation methodology. The literature review found that there were many existing IT and forensic based tools that could be used for USB examinations. The toolset evaluations benchmarked a more manageable sample set of four commonly used tools in New Zealand (namely the *USBDeview*©, USBDeviceForensics©, EnCase® Forensic, and FTK®/FTK® RegistryViewer® tools) against each other. Results from each of the sample toolset evaluations were reported in Chapter 4 and discussed in Section 6.1.1. Overall the USBDevice-Forensics©, EnCase® Forensic and FTK® tools were found to be the most compatible tools for comprehensive analysis and reporting of USB based forensic examinations.

Sub-question 5: What key tool features could be incorporated into the proposed tool design to benefit future USB forensic examinations?

The gap analysis findings presented in Chapter 4 assisted in locating weaknesses in the analytical and reporting features of the sample toolset. A number of existing and new tool features could therefore be improved or incorporated in the design of USBForensicReporter© prototype tool. Potential features included: an automated physical USB device to suspect registry comparative analysis function; automatic report creation to speed up processing times, the ability to easily adapt HTML reports to include specific concepts, processing and legal notes depending on an organisation's reporting requirements or judicial standards and obligations.

Sub-question 6: What protection mechanisms need to be incorporated into the proposed tool design for reliable data output?

A number of built-in protection mechanisms were implemented and discussed in Sections 5.1.1 and 6.1.3 to provide reliable and forensically sound data output. The mechanisms range from the inclusion of testing routines, step-through-analysis and debugging methods in the software coding, read only and independent mounting of suspect registry and system files, and industry standard MD5 hashing algorithms during tool processing. Data output was also verified against the earlier benchmarked data collections and reports to ensure the accuracy and completeness of output be being produced by the *USBForensicReporter*© tool.

Sub-question 7: What improvements does the proposed software reporting tool need to have on existing USB forensic tools and recovery techniques?

The USBForensicReporter© tool interface was found to be easy to use with minimal actions being required of the user to instigate and complete processing and reporting of suspect evidence sets when compared to the manual recovery processes of three of the evaluated tools (namely EnCase® Forensic, USBDeviceForensics© and FTK® RegistryViewer®). Processed information is progressively displayed to the user line-by-line by the tool's user interface as well as being printed to a text-based log file for disclosure purposes. This is not a common feature of the existing tools.

Overall processing results captured during field testing of the developed tool in Table 5.4 indicated that the developed USBForensicReporter© prototype tool analysed and reported USB forensic artifacts on an average of 1.25 seconds whilst utilising the same test datasets as the earlier toolset evaluations. An average processing time of less than 1 ½ seconds gives the tool an exceptional processing ability for future USB forensic examinations. The processing results also demonstrate that automated tools and processes have a clear advantage in performance over manual analysis tools and reporting processes when multiple USB devices are being examined in a case.

The USBForensicReporter© prototype tool also provided improvements to existing analysis methods that are largely based on the individual analysis of evidence files and individual "offline" Registry files, or "live" USB device analysis of a connected USB device. The developed tool instead uses a combination of "live", "offline" Registry files, and the physical USB storage device to conduct a unique comparison of registry and system file data during the processing of suspect evidence sets. The comparative analysis method has been shown in Chapter 5 to accurately make a determination if the physical or suspect USB device has been previously connected to the suspect computer system or not. None of tools in the sample toolset have this level of analytical functionality in their current design features.

6.3.2. The Main Research Question

The main research question was defined in Section 3.2.1 and seeks to answer: **What tool design features improve end-user analysis and reporting of USB forensic artifacts?** The main question and supporting sub-questions have been subsequently tested and answered by utilising a mixed-method research approach. The approach combined elements of qualitative research in the form of toolset evaluations and content analysis, and quantitative research through the use of common forensic tools and laboratory based experiments to provide the basis for a new USB forensic tool. In order for design features to be improved, a baseline of common tool features had to be established in the first instance with a number of existing tools.

The toolset evaluations successfully measured overall tool capability and performance of four commonly used tools against evaluation criteria based on established NIST tool testing methods. The evaluations results also provided a standardised benchmark of performance which had not been published in earlier research studies. The evaluations also allowed the researcher to identify a range of features that each sample tool was capable of employing during USB forensic examinations.

The analysis of results from the Phase Two Gap Analysis presented in Section 4.3.3 identified three significant gaps in analysis and reporting functionality of the evaluated tools. The design of the USBForensicReporter© tool incorporated a number of new and improved features so as to significantly enhance both analytical and output reporting of USB forensic examinations for the end-user. A summary of these specific features is provided as follows:

- The capability to analysis and report on the common Windows® 7 operating system evidence groups identified in Table 3.2 whilst utilising a single tool interface;
- The deployment of a unique comparative-analysis method to associate both the physical USB device and suspect registry files to each other;
- Provision in the tool for both text based log and HTML file formats that are easily adaptable to different organisational reporting standards, and are printable without further formatting by the user;
- A unique *Device Alert Status* field and associated colour-coded reporting panels in the HTML report which establishes if a suspect USB device has or has not previous been attached to the suspect computer system under examination;
- The inclusion of a Notes Section in the HTML report for related Windows® registry concepts and tool processing results to be explained for the benefit of the investigator or client.

The first, second, fourth and fifth feature details are new in the developed prototype tool whilst the third is an improvement to common analysis and reporting features that were found in the existing sample toolset.

All of the features have been shown in development testing to enhance the existing analysis and reporting of USB forensic artifacts whilst utilising the benchmarked toolset evaluation datasets thereby answering the main research question and hypothesis statement.

6.3.3. The Main Hypothesis Test

The main hypothesis statement was developed in Section 3.2.3 and is framed as follows: **USB digital forensic examinations are improved by enhancing the analysis and reporting capability of software tools**. The hypothesis has been tested in the current research by a number of different empirical methods. Methods that were used in the collection and analysis of test data included toolset evaluations, field experiments and the development of a new prototype tool in order to prove or refute the hypothesis. Supporting evidence is also presented to assist in answering the hypothesis statement

Based on the research findings, the main hypothesis is proven and is supported by testing the enhanced analysis and reporting capability of the newly developed USBForensicReporter© prototype tool. USB toolset evaluations were successfully used as a measurement instrument in Section 4.2 to benchmark a common standard of tool functionality and performance which had not been published before. The analysis of the results presented in Section 4.3.3 identified three areas of improvement relating to the analysis and reporting features of the evaluated toolset. New and improved forensic capability was added to future USB forensic examinations by the development of a forensic USB analysis and reporting tool in Chapter 5.

The USBForensicReporter© prototype tool is capable of examining and reporting on all evidence groups found in Table 3.2 that are specially related to the current Windows® 7 operating system. The tool also has the ability to analyse, compare and report on physical USB devices and Windows® Registry files that are in a "live" or "offline" state to accurately determine if the USB device has ever been connected to the computer system under examination. The comparative-analysis design feature is unique to the USBForensicReporter© prototype tool and allows for comprehensive forensic reporting to be conducted by the user.

The current tool design is based on traditional and common industry based software development, design framework and tool testing models (i.e. CLC and RAD software design models, the design science framework and NIST tool testing methodologies). Such a robust software design will allow tool enhancements and support capabilities for other Windows operating system version to be rapidly developed, tested and deployed to the end-user in a final production version of the software.

Currently, the prototype's code support is being expanded to make provision for USB examinations relating to the older Windows® XP operating system as it is still seen in large numbers across New Zealand based laboratories. Future development also includes support for the next version of the Windows® operating system, Windows® 8, which is reported to debut in late 2012.

6.4. CONCLUSION

Chapter 6 has provided a discussion on the overall research findings using key outcomes reported from the sample toolset evaluations in Chapter 4 and the tool development life cycle reported in Chapter 5. The main research question and associated sub-questions proposed in the research methodology of Chapter 3 have been answered in order to test the validity of the research hypothesis.

The findings identified that the hypothesised theory was proven to be a valid statement due largely to the results obtained from testing of the USBForensicReporter© tool. The researcher introduced a number of original analytical and reporting features in order to improve the examination capabilities of existing USB forensic tools.

Field testing and benchmarking against a sample toolset found the USBForensicReporter© tool to be very capable of accurately analysing and reporting data from USB forensics examinations. The tool's general performance was equal to that of several well-known and commercially available forensic tools whilst also surpassing other USB tools that had been tested. Finally, an overall summary of the major research findings is presented in Chapter 7. Future areas of related research will be identified to further build on the body of knowledge within the USB and digital forensics disciplines.

A schedule for the implementation of the USBForensicReporter© tool from a working prototype to final production version will also be outlined in order to conclude the research study.

Chapter 7

Conclusion

7.0 INTRODUCTION

The current research was conducted in order to examine whether USB memory device forensic examinations can be improved by enhancing the analysis and reporting capabilities of existing software tools. A sample set of existing USB forensic and data extraction tools was evaluated in order to derive information need for the development of a working prototype tool for use in future USB forensic examinations. The research will be concluded by providing a summary of the research findings and avenues for further research in the development of the USBForensicReporter© prototype tool and broadening of USB related forensic frameworks.

Chapter 1 introduced the particular research problem and the major components of the study. These components related to development of a literature review, research methodology and discussion of the findings. The first chapter also provided an overview of the main research questions and hypothesis statement.

Chapter 2 provided a contextual basis for the research by exploring a number of specific areas related to USB devices and forensic examinations. These areas included the general use of USB devices, the Windows® registry and USB device connections. Non-windows related USB examinations, USB frameworks and the emergence of anti-forensic techniques were also identified to provide a more in-depth background to the research area. A number of problem areas and issues were identified in relation to the lack of targeted Windows® 7 based forensic examinations, USB memory/storage device frameworks, and toolset assessment in existing research studies.

Chapter 3 developed the research methodology together with the associated research questions and main hypothesis statement. A mixed-method research strategy was adopted. The mixed-method approach used both quantitative, qualitative and software development research elements to fully explore the problem area.

Laboratory and field experiments were therefore successfully employed as a quantitative data gathering strategy to collect object data in the form of USB artifacts for further analysis. A research journal comprising observational notes and output records from the software evaluations and tool development testing was also successfully used as part of the qualitative research element.

In both phases of the quantitive and qualitative research, computer-based forensic software in the form of an existing sample toolset and the developed tool proved to be effective in collecting and analysing data for the research study. The forensic tools allowed data to be selected and retrieved from larger Widows® 7 registry and system files. Selected data was then organised using bookmarking and reporting features of each tool for further analysis. Specific data interpretation features also assisted in data and timestamp interpretation and displayed data in different screen outputs to allow multiple USB artifacts and tool results to be quickly analysed and compared against each other. Overall, the software output helped the researcher in drawing conclusions about the findings of each tool evaluation cycle in order to test the hypothesis and find answers for each of the research sub-questions.

Chapter 4 reported the findings for the sample toolset evaluations and gap analysis. The toolset evaluation results provided a new and unpublished benchmark of toolset performance for a sample set of tools that are currently used by New Zealand based digital forensic laboratories. The gap analysis method proved its value by displaying the Phase One benchmark results while also identifying the presence of potential analysis and reporting performance gaps in the Phase Two results. Performance gaps allowed for new features and improvements to be made in the tool design process of the USBForensicReporter© tool.

Chapter 5 outlined the development of the USBForensicReporter© tool including a number of design enhancements to its analytical and reporting abilities. Established tool validation and data verification methods were used in field testing of the prototype tool. The field results determined the prototype tool was capable of analysing and reporting USB devices and forensic artifacts promptly and to the standard of existing commercial tools utilised during the toolset evaluations and benchmarking phases of the study.

Chapter 6 discussed important elements of the research study and provided a summary of each of the research phases relating to the toolset evaluations, Gap Analysis, and development of the USBForensicReporter© prototype tool. A number of limitations relating to the toolset evaluations, selection of USB devices and software design were also discussed before the research questions and hypothesis were answered.

7.1. SUMMARY OF FINDINGS

The research findings are based on the results of the sample toolset evaluations and Gap Analysis presented in Chapter 4 together with the results of the tool development and field testing presented in Chapter 5. The toolset evaluations employed a modified version of established NIST CFTT tool testing methods to benchmark performance and provide a comparison of overall tool capability. The Gap Analysis measured conformance to a specific testing criteria whilst also identifying significant weaknesses in tool operation and functionality in order to determine what new or improved features needed to be included in the proposed tool design.

Four USB software tools were evaluated as part of the sample toolset evaluations. Three of the tools were forensic-based whilst the other (USBDeview©) can best be described as a USB software utility. Six USB based memory storage devices supporting the more common USB 2.0 and the newer USB 3.0 specifications were also utilised during the evaluations to provide a wider range of thumb drive and PSD device information for data collection purposes. All of the USB 2.0 supported thumb drives and PSD devices were able to be tested with varying degrees of data output during Tests 1 to 7. The Test 8 evaluations found that each of the tools except the USBDeview© software utility were capable of analysing and reporting on USB devices that supported the latest USB 3.0 specification.

The evaluation findings showed the FTK®/FTK® RegistryViewer® and EnCase® Forensic tools performed the best at providing a comprehensive forensic analysis and reporting of USB artifacts for both USB 2.0 and 3.0 supported memory storage devices.

The *USBDeviceForensics*© tool was a close second but lacked support for examining an important subkey in the *SOFTWARE* registry hive file. The unsupported *Windows Portable Devices* subkey caused past drive letter assignments not to be captured for the USB thumb drives used in Tests 1 to 4. The research findings also determined that the *Windows Portable Devices* subkey only records device information and previous drive letter assignments for USB thumb drives and not PSD devices with conventional hard drives. Drive letter assignments for PSD devices are not recorded in the *Windows Portable Devices* subkey, only in the *MountedDevices* subkey of the *SYSTEM* registry hive file.

The USBDeview© tool was found to be best suited for general IT system administration tasks involving the identification of connected USB devices on "live" computer systems, and auditing of extracted *SYSTEM* hive files via the command-line analysis option. From a forensic perspective, only *device descriptor* information from the USB device and *SYSTEM* registry hive file information could be collected by the tool's current design, limiting the tool's overall collection capability. The other tools in the sample toolset provided higher levels of evidence collection and more in-depth methods of USB forensic examination than the USBDeview© tool.

The gap analysis method presented in Figure 4.6 proved a viable assessment instrument for evaluating performance, operability and functionality in forensic based software tools. The categorisation results in Table 4.6 and the Microsoft® radar mapping method illustrated in Figure 4.7 successfully identified three significant weaknesses in analysis and reporting functionality of the toolset sample. The findings from the gap analysis method enabled improvements to be made to the analysis and reporting functions of the developed prototype tool. The process also allowed the researcher to develop new and creative features in tool design to further enhance USB based forensic examinations.

The proposed research methodology in Chapter 3 was built upon established design science frameworks, software design models and tool testing frameworks. All the five phases of the adopted research methodology were successfully implemented with only coding issues being encountered during the software design phase in order to produce a new and working USB analysis and reporting tool.

The findings from the prototype tool development and field testing identified that the USBForensicReporter© tool in its current working prototype version was stable during its operation. The tool was also found to be more than capable of accurately analysing and reporting both physical USB storage devices and Windows® 7 operating system artifacts associated to USB memory device forensic examinations.

The prototype tool design provides a new and innovative comparability based function that allows the user to exam both physical USB storage devices and "offline" suspect registry evidence sets at one time by interacting with a single tool interface. The physical linking of USB devices to a computer system under investigation is a critical requirement of most USB forensic examinations. None of the tools in the sample toolset have this level of analytical functionality and comparability between both types of evidence sets in their current design states. The prototype tool also improves on existing forensic reporting standards by providing standardised, editable and comparison based evidence reporting in an HTML format and full text-based process logging for verification and disclosure purposes.

7.2. SUMMARY OF RESEARCH QUESTIONS AND HYPOTHESIS

The main hypothesis statement and associated research questions developed in Chapter 3 were derived from gaps found in the literature review presented in Chapter 2. In order for the hypothesis and research questions to be answered, empirical and experimental research in the form of toolset evaluations, data analysis and software tool development was undertaken. The results from each of the different research stages were summarised in Chapters 4 and 5 and the research findings were discussed in Chapter 6. Answers to each of the research sub-questions form the basis for answering the main research question and hypothesis statement, and are ordered in Table 7.1 to reflect this logical progression.

Table 7.1

Research Questions and Answers Summary

Research Questions	**Answers**
Sub-Question 1: What is the current state of forensic research related to USB storage devices?	**A:** The literature review in Chapter 2 found past USB research had largely centred on Windows XP operating system artifacts. However, Windows® 7, Macintosh® and Linux® operating systems research is growing in popularity as new operating system platforms are released. The ubiquitous nature of USB technology has meant more emphasis is being placed on these types of devices in both criminal and corporate investigations.
Sub-Question 2: What operating system records are generated by USB activity on a Windows computer system?	**A.** Operating system records are generated in various *SYSTEM*, *SOFTWARE*, *NTUSER.DAT* related registry hive files and in system logs such as the *setupapi.dev.log* and setupapi.log files for USB related activity.
Sub-Question 3: What specific Windows 7 Registry evidential related artifacts can assist a forensic practitioner in USB examinations?	**A.** Information contained in the *Windows Portable Devices* sub-key of the *SOFTWARE* registry hive can assist the examiner in determining USB thumb drive serial numbers and recent drive letter mappings if the drive letter in the *MountedDevices* sub-key of the *SYSTEM* registry hive has been reallocated to another USB device connection. The *USBSTOR* subkey of the *SYSTEM* registry hive provides a comprehensive list of past USB device activity and device information. Further USB device information and connection dates and times were also located in the *setupapi.dev.log* system file to identify the first date and time that a particular USB device was connected to a Windows® 7 based computer system.
Sub-Question 4: What forensic or commercial tool examples are currently available to the examiner for collecting and reporting on USB artifacts?	**A.** The literature review found that there were many existing IT and forensic based tools that could be used for USB examinations. Only a sample set of four commonly used tools were evaluated during the current research due to time and resource constraints. The freeware examples included the USBDeview© and USBDeviceForensics© tools whilst the commercial examples included the EnCase® Forensic and FTK®/FTK RegistryViewer® toolsets.
Sub-Question 5: What key tool features could be incorporated into the proposed tool design to benefit future USB forensic examinations?	**A.** Both the sample toolset evaluations and Gap Analysis findings determined the following features would be beneficial to further USB examinations: **1.** Full automation of all evidence mounting and analysis functions. **2.** Physical USB device to evidence set comparability analysis within a single tool interface. **3.** Full automation of both HTML reporting and text based logging outputs. **4.** Standardised reporting outputs that are easily adaptable to organisational reporting requirements and judicial obligations.

Sub-Question 6: What protection mechanisms need to be incorporated into the proposed tool design for reliable data output?	**A.** The protection mechanisms in the prototype tool design range from the inclusion of testing routines, step-through-analysis and debugging methods in the software coding, read only and independent mounting of suspect registry and system files, and industry standard MD5 hashing of evidence set files during tool processing to maintain reliability of data output.
Sub-Question 7: What improvements does the proposed software reporting tool need to have on existing USB forensic tools and recovery techniques?	**A.** The developed USBForensicReporter© tool provides a single forensic software tool interface that conducts comparability analysis and reporting on physical USB devices and suspect evidence sets at the same time. **1.** The automated analysis and reporting features of the tool considerably decrease the overall processing and reporting times when compared against manual processing functions of existing forensic based tools such as EnCase® Forensic, FTK® RegistryViewer® and USBDeviceForensics©. **2.** Past and specific information related to all USB device references contained within the *USBSTOR* subkey of the suspect *SYSTEM* registry hive file are automatically reported in the HTML report format. **3.** A unique red coloured panel scheme and "Device Alert Status" indicators are also utilised in the HTML report to quickly identify if a suspect USB device has previously been connected to the suspect computer system under examination. None of the existing tools have these unique analytical and reporting features at the current time.
Main Research Question: What tool design features improve end-user analysis and reporting of USB forensic artifacts?	**A:** The research findings established the following features improve end-user USB forensic analysis and reporting: **1.** The inclusion of a physical USB device and registry file comparability analysis method to evidentially associate USB devices to a suspect computer system using only one tool interface. **2.** The capability to analyse and report on all common Windows® 7 operating system evidence groups with one tool interface only. **3.** The provision for both text based log and HTML file formats that are easily adaptable to different organisation reporting standards, and can be easily printed without further formatting by the user. **4.** The inclusion of a "Notes Section" in tool output to explain USB or system concepts and tool processing results for the benefit of the investigator or for disclosure purposes.
The Main Hypothesis Statement: USB digital forensics examinations are improved by enhancing the reporting capability of software tools	The hypothesis is **proven** by successfully testing the enhanced analytical and reporting capability of the newly developed USBForensicReporter© tool against a benchmarked sample set of existing USB and forensic analysis tools.

7.3. FUTURE RESEARCH

A number of areas of future research have been identified as a result of the current research study. The areas are largely based around the USBForensicReporter© tool, USB memory/storage device frameworks and the need for further USB forensic tools to be evaluated in order to create a wider and more in-depth benchmarking program. Advancement of the prototype's overall development and broadening of the toolset benchmarking program will continue throughout 2012. However, the implementation of a dedicated USB examination framework that is both scientifically sound and tool neutral will take a longer time to implement as further research needs to be conducted on integrating methods used in the current research with other established forensic framework components.

The research findings in this study have shown that the USBForensicReporter© tool developed as a "proof of concept" tool, is capable of producing detailed and reliable USB related forensic evidence; however, there are still a number of features that are currently being refined or need further enhancement in order for the tool to be delivered as a full production version. Since November 2011, small incremental changes have been made to the tool's processing and reporting code to produce a more advanced and consistent standard of reporting. Likewise, some small data value anomalies that specifically related to the display of PSD drive signatures and the looping of drive letter assignments required further testing and refinement to resolve these issues.

As of January 2012, some additional code samples were developed and tested to provide a more robust way of mounting the suspect registry files than previously designed. The support for Window® XP related artifacts has also been incorporated into the tool's coding (further experimental testing is still to be completed) in order to provide greater support for common Windows® operating systems that are likely to be encountered in a digital forensics laboratory. The researcher would also like to provide future tool support for the next generation of the Windows® operating system, Windows® 8 once an exact release date has been published by Microsoft®.

On a Windows® Registry level, the *SOFTWARE\Microsoft\Windows NT\ CurrentVersion\EMDMgmt* subkey needs further research before the associated information can be analysed and reported by the prototype tool. Preliminary and limited testing results proved to be inconclusive in determining consistency of changes to the *LastTestedTime* date and time stamp associated to this subkey. Additional hypothesis and experimental testing is also required by the researcher to make a determination if the *LastTestedTime* date and time stamp value could be used as a fundamental forensic artifact reference in establishing when a USB storage device was last connected to Windows® 7 based computer system.

References

AccessData. (2011). *FTK overview*. Retrieved February 06, 2011, from http://accessdata.com/products/computer-forensics/ftk

ACPO. (2007). *Good practice guide for computer-based electronic evidence: Retrieval of video & CCTV evidence*. Retrieved from http://www.7safe.com/electronic_evidence/ACPO_guidelines_computer_evidence.pdf

Alghafli, K.A., Jones, J., Martin, T. A. (2010). *Forensics analysis of the Windows 7 Registry*. Retrieved from http://igneous.scis.ecu.edu.au/proceedings/2010/adf/Alg.pdf

Altheide, C., & Casey, E. (2010). UNIX forensic analysis – Removable media. In E. Casey (Ed.), *Handbook of digital forensics and investigations 2010* (p.338). Burlington, MA: Elsevier Academic Press. doi: 10.1016/B978-0-12-374267-4.00013-6

Al-Zarouni, M., & Al-Hajri, H. (2007). A proof-of-concept project for utilizing U3 technology in incident response. *Proceedings of the 5th Australian Digital Forensics Conference.* Edith Cowan University, Perth Western Australia. Retrieved October 04, 2010, from http://ro.ecu.edu.au/adf/15/

Amaral, L.M.G., Faria, J.P. (2010). A gap analysis methodology for the team software process. *Proceedings of Quality of Information and Communications Technology (QUATIC), 2010 Seventh International Conference on the*, vol., no., 424-429. doi: 10.1109/QUATIC.2010.78

Andrew, Michael. W. (2007). Defining a process model for forensic analysis of digital devices and storage media, *Systematic Approaches to Digital Forensic Engineering, 2007. SADFE 2007. Proceedings of the Second International Workshop on* , vol., no., 16-30. doi:10.1109/SADE.2007.8

Andriessen, D. (2007). *Combining design-based research and action research to test management solutions*. Paper presented at the 7th World Congress Action Research, Groningen, The Netherlands. Retrieved from http://www.weightlesswealth.com/downloads/Andriessen%20DBR%20and%20Action%20Research.pdf

Ayers, D. (2009). A second generation computer forensic analysis system. *Digital Investigation*, 6(S), 34-42. doi: 10.1016/j.diin.2009.06.013

BBC. (2009). *Previous cases of missing data*. Retrieved February 04, 2011, from http:// news.bbc.co.uk/2/hi/uk_news/7449927.stm

Beckett, J., & Slay, J. (2007). Digital forensics: Validation and verification in a dynamic work environment. *System Sciences, 2007. HICSS 2007. In the proceedings of 40th Annual Hawaii International Conference on* , vol., no., pp.266a, Big Island, Hawaii. doi: 10.1109/HICSS.2007.175

Beebe, N. L. (2009). Digital forensic research: The good, the bad, and the unaddressed. In V, G. Peterson and S. Shenoi (Eds.), *Advances in digital forensics* (pp.17-33). Boston, MA: Springer

Bell, G.B., & Boddington, R. (2010). Solid state drives: The beginning of the end for current practice in digital forensic recovery? *JDFSL, The Journal of Digital Forensics, Security and Law*, 5(3), 1-20. Retrieved from http://www.jdfsl.org/subscriptions/JDFSL-V5N3-Bell.pdf

BlackBag Technologies. (2011). *Snow leopard logs USB serial numbers*. Retrieved February, 03, 2010, from https://www.blackbagtech.com/blog/category/mac-forensics/

Bosschert, T. (2006). Battling anti-forensics: Beating the U3 stick, *Journal of Digital Forensic Practice*, 1: 4, 265 — 273. Retrieved October 01, 2010, from http://www.tandf.co.uk/15567281. doi: 10.1080/15567280701417975

Breeuwsma, M., de Jongh, M., Klaver, C., van der Knijff, R., & Roeloffs, M. (2007). Forensic data recovery from flash memory. *Small Scale Digital Forensics Journal*, VOL. 1, NO. 1, JUNE 2007, 1-17. Retrieved from http://www.ssddfj.org/papers/SSDDFJ_V1_1_Breeuwsma_et_al.pdf

Carrier, B. (2006). *A hypothesis-based approach to digital forensic investigations*. CERIAS tech Report 2006-06. Retrieved from Purdue University, The Centre for Education and Research in Information Assurance and Security website: https://www.cerias.purdue.edu/assets/pdf/bibtex_archive/2006-06.pdf

Carvey, H., & Altheide, C. (2005). Tracking USB storage: Analysis of windows artifacts generated by USB storage devices. *Digital Investigation (2005)*, 2, 94-100. Burlington, MA: Elsevier Academic Press. Retrieved from http://gs.hui.edu.vn:8000/collect/EN-digital/index/assoc/HASH2a62.dir/1c(64).pdf

Carvey, H. (2009). Registry Analysis, *Windows forensic analysis DVD toolkit 2E* (pp. 206-218). Burlington, MA: Elsevier.

Carvey, H. (2011). *Windows registry forensics: Advanced digital forensic analysis of the windows registry*. Burlington, MA: Elsevier.

Casey, E. (2010). *Handbook of digital forensics and investigations 2010.* In E. Casey (Ed.), (p.2). Burlington, MA: Elsevier Academic Press.

Casey, E. (2011). Computer basics for digital investigators. In E.Casey (ed.), *Digital evidence and computer crime* (3rd ed., p.442). Waltham, MA: Elsevier Academic Press

Craig, R.D. & Jaskiel, S.P. (2002). *Systematic software testing*. Boston, MA: Artech House Publishers

Cosic, J., & Baca, M. (2010. (Im)proving chain of custody and digital evidence integrity with time stamp. *MIPRO, 2010 Proceedings of the 33rd International Convention*, vol., no., 1226-1230.

DataLossDB. (2011). *Incident 2627*. Retrieved February, 04, 2011, from http://datalossdb.org/incidents/2627-data-of-3-3m-names-addresses-dates-of-birth-and-social-security-numbers-exposed-on-stolen-portable-media-device

DeAbren, B. (2000). Test software design techniques for reuse and portability, *AUTOTESTCON. Proceedings of 2000 IEEE*, vol., no., 334-338. doi:10.1109/AUTEST.2000.885610

DFRWS. (2001). Report from the First Digital Forensic Research Workshop (DFRWS). *DTR - T001-01 FINAL*. Utica, NY. Retrieved from DFRWS website: https://www.dfrws.org/2001/dfrws-rm-final.pdf

Erbacher, R.F. (2010). Validation for digital forensics. In the proceedings of *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on* , vol., no., pp.756-761, Las Vegas, NV. doi: 10.1109/ITNG.2010.18

Farmer, D. (2007). A forensic analysis of the Windows Registry. Retrieved from http://www.eptuners.com/forensics/contents/A_Forensic_Examination_of_the_Windows_Registry_DETAILED.pdf

Fisher, M.S. (2007). *Software verification and validation: An engineering and scientific approach*. New York, NY: Springer Science+Business Media

Gao, Q., & Wu, S. (2009). Research of recycle bin forensic analysis platform based on XML techniques, *Software Engineering, 2009. WCSE '09*. WRI World Congress on, vol.4, no., pp.337-341, 19-21. doi:10.1109/WCSE.2009.111

Gao, Y. (2010). Research on the rule of evolution of software development process model, Information Management and Engineering (ICIME), 2010. *Proceedings of the 2nd IEEE International Conference* on , vol., no., pp.466-470, 16-18. doi:10.1109/ICIME.2010.5477884

Gorge, M. (2005). USB & other portable storage device usage: Be aware of the risks to your corporate data in order to take pre-emptive and/or corrective action, *Computer Fraud & Security*, 2005(8), 15-17. doi: 10.1016/S1361-3723(05)70244-X

Guidance Software. (2011). *EnCase*® Forensic. Retrieved July 25, 2011, from http://www.guidancesoftware.com/forensic.htm

Guo, Y., Slay, J., & Beckett, J. (2009). Validation and verification of computer forensic software tools – Searching function. *Digital Investigation*, 6, S12-S22.

Harvey, D., & Ayers, D. (2010). *Computer Forensics – using and testing expert computer evidence*. (p.74). Auckland, New Zealand : NZLS CLE Ltd

Hevner, A. R., March, S.T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105. Retrieved October 05, 2010, from http://www.jstor.org/pss/25148625

Hoffman, R. R., & Deal, S. V. (2008). Influencing versus informing design, Part 1: A gap analysis. *IEEE: Intelligent Systems*, 23(5), 72-75. doi: 10.1109/MIS.2008.83

Honeycutt, J. (2003). *Windows XP registry guide*. Redmond: WA: Microsoft Press.

IEEE. (2005). IEEE standard for software verification and validation, *IEEE Std 1012-2004*, (p.9). New York, NY: Institute of Electrical and Electronics Engineers. Retrieved December 18, 2001, from http://ieeexplore.ieee.org.

Iivari, J. (2007). A paradigmatic analysis of information systems as a design science, *Scandinavian Journal of Information Systems*, 19(2), 39-64. Retrieved from http://www.hec.unil.ch/yp/DRIS/Articles/Iivari07.pdf

Intel. (2004). Endianness white paper. Retrieved from http://www.intel.com/design/intarch/papers/endian.pdf

ISTS. (2004). Law enforcement tools and technologies for investigating cyber attacks: Gap analysis report. Retrieved from Dartmouth College Institute for Security, Technology, and society website: http://www.ists.dartmouth.edu/docs/ISTSGapAnalysis2004.pdf

Jalote, P. (2005). Software requirements analysis and specification. In D.Gries & F.B. Schneider (Eds.), *An integrated approach to software engineering* (3rd ed., pp.91-99). New York, NY: Springer Science + Business Media.

Jones, A., Valli, C., & Dabibi, G. (2009). The 2009 analysis of information remaining on USB storage devices offered for sale on the second hand market. *Proceedings of the 7th Australian Digital Forensics Conference 2009*, 7-13. Perth, Australia: Edith Cowan University. Retrieved from http://igneous.scis.ecu.edu.au/proceedings/2009/forensics/Jones_Valli_Dabibi.pdf

Kemble, A. (2008). *Forensic computing: use of Linux log data in USB portable storage device artefact analysis*. (Masters Dissertation). Open University, Milton Keyes, United Kingdom. Retrieved from http://computing-reports.open.ac.uk/2008/TR2008-24.pdf

Kokocinski, A. (2010). Macintosh forensic analysis. In E. Casey (Ed.), *Handbook of digital forensics and investigations 2010* (pp. 368-369). Burlington, MA: Elsevier Academic Press. doi: 10.1016/B978-0-12-374267-4.00013-6

Lee, K., Lee, W., Park, C., Bang, J., Kim, K., & Lee, S. (2008). USB Passon: Secure USB thumb drive forensic toolkit, fgcn, vol. 2, pp.279-282, 2008 *Second International Conference on Future Generation Communication and Networking*. doi: 10.1109/FGCN.2008.214

Lee, R. (2009). *Computer forensic guide to profiling USB device thumbdrives on Win7, Vista, and XP*. Retrieved October 01, 2010, from http://blogs.sans.org/computer-forensics/2009/09/09/computer-forensic-guide-to-profiling-usb-thumbdrives-on-win7-vista-and-xp/

Lehman, T.J., & Sharma, A. (2011). Software development as a service: Agile Experiences. *Proceedings of the SRII Global Conference (SRII),* 2011 Annual , vol., no., pp.749-758. doi: 10.1109/SRII.2011.82

Liu, X. (2008). *Bluetooth information exchange network.* (Master's thesis, Auckland University of Technology, Auckland, New Zealand). Retrieved from http://aut.researchgateway.ac.nz/bitstream/handle/10292/7 22/LiuX.pdf?sequence=4

Luo, V. (2007). Tracing USB device artifacts on Windows XP operating system for forensic purpose. In the Proceedings of the *5th Australian Digital Forensics Conference 2007*, 210-218. Perth, Australia: Edith Cowan University. Retrieved from http://scissec.scis.ecu.edu.au/proceedings/2007/ forensics/23_Luo_Tracing_USB_Device_artefacts_on_Windows_XP.pdf

Paramount Software. (2011). *Macrium reflect standard v5*. Retrieved March, 15, 2011, from http://www.macrium.com/personal.aspx?FREELINK=Y

March, S. T., & G. F. Smith. (1995). Design and natural science research on information technology. *Decision Support Systems* 15: 251-266. doi: 10.1016/0167-9236(94)00041-2

Menz, M., & Bress, S. (2004). *The fallacy of software write protection in computer forensics*. Retrieved from http://www.mykeytech.com/ SoftwareWriteBlocking2-4.pdf

Microsoft (2003). *Troubleshooting device installation with the SetupAPI Log file.* Retrieved February 04, 2011, from http://www.microsoft.com/whdc/ archive/setupapilog.mspx

Microsoft. (2010). *Standard USB identifiers*. Retrieved February 04, 2011, from http://msdn.microsfot.com/en-us/library/ff553356

Microsoft. (2011a). *Windows 7, Windows Server 2008 R2, and Windows Vista setup log file locations*. Retrieved January 29, 2010, from http://support.microsoft.com/kb/927521

Microsoft. (2011b). *Windows registry information for advanced users*. Retrieved January 29, 2010, from http://support.microsoft.com/kb/256986

Microsoft. (2011c). *File times*. Retrieved December 18, 2011, from http://msdn.microsoft.com/en-us/library/windows/desktop/ms724290(v=vs.85).aspx

Mingers, J. (2001). Combining IS research: Towards a pluralist methodology, *Information Systems Research*, 12(3), 240-259. Retrieved from http://155.97.60.20/7910F/papers/ISR%20combining%20IS%20research%20methods.pdf

Mokube, I. (2008). Digital forensics: forensic analysis of an iPod shuffle. *Proceedings of the 46th Annual Southeast Regional Conference* on XX (Auburn, Alabama, March 28 - 29, 2008). ACM-SE 46. ACM, New York, NY, 215-219. doi: 10.1145/1593105.1593161

Mueller, S. (2011). *Upgrading and repairing PCs 20th Edition: Magnetic storage*. Indianapolis, ID: Que Publishing

Nance, K., Hay, B., & Bishop, B. (2009). *Digital forensics: defining a research agenda*. Retrieved from http://assert.uaf.edu/papers/dfResearchAgenda_HICSS09.pdf

Net applications. (2011). *Operating system market share - December 2010*. Retrieved January 30, 2011, from http://www.netmarketshare.com/report.aspx?qprid=10&qptimeframe=M&qpsp=148

Net applications. (2011a). *Operating system market share - January 2011*. Retrieved January 30, 2011, from http://marketshare.hitslink.com/operating system-market-share.aspx?qprid=10. January 30 2011.

New Zealand Government. (2010). Evidence Act 2006 No 69 (as at 07 July 2010), Public Act: *Section 137 evidence produced by machine, device, or technical process*. Retrieved December 04, 2011, from http://www.legislation.govt.nz/act/public/2006/0069/latest/DLM393979.html

NirSoft. (2006-2011). *USBDeview v1.86*. Retrieved October 20, 2010, from http://www.nirsoft.net/utils/usb_devices_view.html

NIST. (2002). *Setup and test procedures dd (GNU fileutils) 4.0.36 forensic tests Version 1.0, (37-38).* Gaithersburg, MD: National Institute of Standards and Technology, U.S. Department of Commerce. Retrieved from NIST website: www.cftt.nist.gov/setup_for_dd_tests.pdf

NIST. (2005). *Digital data acquisition tool test assertions and test plan*. Gaithersburg, MD: National Institute of Standards and Technology, U.S. Department of Commerce. Retrieved from NIST website: http://www.cftt.nist.gov/DA-ATP-pc-01.pdf

NIST. (2006). *Guide to integrating forensic techniques into incident response* (Special Publication 800-86). Retrieved from NIST website: http://csrc.nist.gov/publications/nistpubs/800-86/SP800-86.pdf

NIST. (2009). *The System development life cycle (SDLC)*. Gaithersburg, MD: National Institute of Standards and Technology, U.S. Department of Commerce. Retrieved from NIST website: http://csrc.nist.gov/publications/ nistbul/april2009_system-development-life-cycle.pdf

NIST. (2011). *CFTT project overview*. Retrieved December 15, 2011, from http://www.cftt.nist.gov/project_overview.htm

Norris, P. (2009). *The internal structure of the windows registry* (Master's thesis, Cranfield University, United Kingdom). Retrieved from http://amnesia. gtisc.gatech.edu/~moyix/suzibandit.ltd.uk/MSc/

Olsson, J., & Boldt, M. (2009). Computer forensic timeline visualization tool, *Digital Investigation*, 6(S), September 2009, S78-S87. *doi:10.1016/j.diin.2009.06.008*

Oxford University Press. (2012) *Definition: forensic*. Retrieved January 1, 2012, from http://oxforddictionaries.com/definition/?q=forensics#forensic__4

Pangalos, G., Ilioudis, C., & Pagkalos, I. (2010). The Importance of Corporate Forensic Readiness in the Information Security Framework," *Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE). Proceedings of 19th IEEE International Workshop on* , vol., no., p.12-16. doi: 10.1109/WETICE.2010.57

Peisert, S., Bishop, M., & Marzullo, K. (2008). Computer forensics in forensic, Systematic Approaches to Digital Forensic Engineering. *SADFE '08*. Third International Workshop on , vol., no., 102-122. doi: 10.1109/ SADFE.2008.18

Pittman, R. D., & Shaver, D. (2010). Windows forensic analysis. In E. Casey (Ed.), *Handbook of digital forensics and investigations 2010* (pp. 209-300). Burlington, MA: Elsevier Academic Press. doi: 10.1016/B978-0-12-374267-4.00013-6

Pressman, R. S. (2001). *Software engineering: A practitioner's approach*. Retrieved from http://ce.sharif.ir/courses/84-85/1/ce474/resources/root/ Pressman_Sofware%20Engineering.pdf

Privacy Commissioner. (2010). *Portable storage device survey: May 2010: New Zealand Public Sector (P/0198/A223526)*. Retrieved February 04, 2011, from http://privacy.org.nz/assets/Files/Surveys/Portable-Storage-Device-Survey-Report-2010.pdf

Quinn, K. J. Spike. (2010). *2010 New Zealand computer crime and security survey*. University of Otago Security Research Group: Dunedin, New Zealand. Retrieved February 05, 2011, from http://eprints.otago.ac.nz/1017/

Quirk, G. (2011). *80 million superspeed devices in 2011*. Retrieved February 08, 2012, from http://nz.mouser.com/applications/usb30_market_potential/

Ryan, D.J., & Shpantzer, G. (2005). *Legal aspects of digital forensics*. Retrieved from http://euro.ecom.cmu.edu/program/law/08-732/Evidence/ RyanShpantzer.pdf

RegRipper. (2011). *RegRipper*. Retrieved October 20, 2011, from http://regripper.wordpress.com/

Richard III, G.G., & Roussev, V. (2006). Digital forensic tools: Next-generation digital forensics, *Communications of the ACM*, 49(2), 77. doi:10.1145/1113034.1113074

Ruparelia, N.B. (2010). Software development lifecycle models. *SIGSOFT Softw. Eng. Notes*, 35(3), 8-13. doi:10.1145/1764810.1764814

Sansurooah, K. (2009). A forensics overview and analysis of USB flash memory devices. *Proceedings of the 7th Australian Digital Forensics Conference 2009*, 99-108. Perth, Australia: Edith Cowan University. Retrieved October, 01, 2010, from http://ro.ecu.edu.au/adf/70/

Slade, M. (2011). *Data theft a growing problem.* Retrieved February 18, 2011, from http://www.nzherald.co.nz/maria-slade/news/article.cfm?a_id=358&objectid=10701468

Sommer, P. (2010). *Forensic science standards in fast-changing environments*. Retrieved from http://oro.open.ac.uk/19455/1/Forensic_Science_Standards_in_Fast_Changing_Environments_S%E2%80%A6.pdf

Spenser, M. (2010). *Most accepted software development models*. Retrieved October 10, 2011, from http://www.plaveb.com/blog/most-accepted-software-development-models

SWGDE. (2006). *SWGDE best practises for computer forensics v2.1*. Retrieved October 04, 2011, from http://www.swgde.org/documents/current-documents/

SWGDE. (2009). *SWGDE recommended guidelines for validation testing version 1.1*. Retrieved from http://www.swgde.org/documents/current-documents/2009-01-15 SWGDE Recommendations for Validation Testing Version v1.1.pdf

SWGDE. (2011). *SWGDE and SWGIT digital & multimedia evidence glossary*. Retrieved from http://www.swgde.org/documents/archived-documents/2011-01-14 SWGDE-SWGIT Glossary v2 4.pdf

TechTarget. (2006). *Definition: Gap analysis*. Retrieved August 29, 2011, from http://searchcio-midmarket.techtarget.com/definition/gap-analysis

The Apple Examiner. (2011). *Initial data gathering*. Retrieved September, 26, 2011, from http://www.appleexaminer.com/MacsAndOS/Analysis/InitialDataGathering/InitialDataGathering.html

Thomas, P., & Morris, A. (2008). An investigation into the development of an anti-forensic tool to obscure USB flash drive device information on a Windows XP platform. *Proceedings of the 2008 Third international Annual Workshop on Digital Forensics and incident Analysis*. *WDFIA*. IEEE Computer Society, Washington, DC, 60-66. doi:10.1109/WDFIA.2008.13

Thomassen, J. (2008). *Forensic analysis of unallocated space in WINDOWS registry hive files* (Master's thesis, The University of Liverpool, United Kingdom). Retrieved from http://sentinelchicken.com/data/JolantaThomassen DISSERTATION.pdf

Toshiba. (2008). *Flash memory fact sheet - memory business unit overview*. Retrieved January 29, 2010, from http://www.toshiba.com/taec/news/media_resources/docs/FlashFactSheet.pdf

Turnbull, B. (2007). *Wi-Fi as Electronic Evidence: Policy, Process and Tools*. (Doctoral thesis, University of South Australia, Adelaide, Australia). Retrieved from http://ura.unisa.edu.au/R/-?func=dbin-jump-full&object_id=unisa44383

USB Implementers Forum. (2000). *Universal serial bus revision 2.0: Background*. Retrieved, from http://www.usb.org/developers/docs/usb_20.pdf

USB Implementers Forum. (2008). *Universal serial bus 3.0 specification: Get descriptor*. Retrieved from http://www.usb.org/developers/docs/ USB 3 0.pdf

USB Implementers Forum. (2009). *USB class codes - base class 08h (mass storage)*. Retrieved February 20, 2011, from http://www.usb.org/developers/ defined_class

USBlyzer.com. (2011). *USBlyzer – USB protocol analyzer and USB traffic sniffer*. Retrieved October 18, 2011, from http://www.usblyzer.com/

Vacca, J., & Rudolph, K. (2011). *System forensics, investigation, and response: Forensic methods and labs*. Sudbury, MA: Jones & Bartlett Learning

van der Knijff, R. (2010). *Handbook of digital forensics and investigations 2010.* In E. Casey (Ed.). Burlington, MA: Elsevier Academic Press.

Venable, J. (2006). Role of theory and theorising in design science research. *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology*, DESRIST 2006, Claremont, CA, pp.1-18. Doi: 10.1.1.110.2475

Woanware. (2011). *USBDeviceForensics*. Retrieved October 20, 2011, from http://www.woanware.co.uk/?page_id=45

Xbit Laboratories. (2010). *USB 3.0: Theory and practice*. *Page 2*. Retrieved February, 09, 2011, from http://www.xbitlabs.com/articles/storage/display/ usb-3_2.html

Yasin, M., Cheema, A. R., Kausar, F. (2010). Analysis of internet download manager for collection of digital forensic artefacts, *Digital Investigation*, 7(1-2), 90-94. doi: 10.1016/j.diin.2010.08.005.

Yun, S., Savoldi, A., Gubian, P., Kim, Y., Seokhee Lee., Sangjin Lee. (2008). Design and implementation of a tool for system restore point analysis. *Proceedings of the Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP '08 International Conference on* , vol., no., pp.542-546. doi: 10.1109/IIH-MSP.2008.256

Appendix A – Research Definitions

Artifact – data related to system and or user activity that is focused on data storage media.

Digital Forensics Practitioner – a trained and qualified person actively engaged in the digital forensics profession or related industries.

Endianess – the order in which bytes are stored in the computer's memory. The Endianess type is defined by the CPU architecture as being either B*ig-endian* or *Little-endian*. In *Big-endian* the most significant bytes or the "big end" are stored first on the left-hand side. In *Little-endian* the least significant bytes or the "little end" are stored first as the most significant bytes are last or at the furthest right-hand side. Modern Intel-based computers are classed as *Little-endian* systems.

Master Boot Record (MBR) – enables the computer system to identify the bootable partition with the operating system and other miscellaneous information on a hard drive.

Software – written programs and instruction code (both system and application based) pertaining to the operation of a computer system that are utilised by users to complete certain tasks.

Storage Media – any physical storage media or device on which data is stored such as conventional hard drives, USB thumb drives and portable storage devices.

Suspect Evidence – In the context of the current research study, the term suspect relates to a common set of Windows® associated files, computer systems or USB storage devices being examined by a digital forensics practitioner for digital evidence. The term is inherently linked in law enforcement terms to a person suspected of engaging in a crime or criminal activity.

Appendix B – Research Journal

Date	Task	Actions, Results and General Observations
24/06/2011	Create Windows® 7 related testing environments using VMware® Workstation (v6.5.5) virtualisation software	Windows® 7 VM Baseline created without issue. Install *USBDeview*© (v1.91) and *USBDeviceForensics*© (v1.0.7) software on the desktop. Create an additional nine templates labelled as "Initial Test" and "Test 1" to "Test 8" from the VM baseline and place into separate test folders.
26/06/2011	Initial blind testing of *USBDeview* and *USBDeviceForensics* software with the VM labelled as "Initial Test".	*USBDeview*© – Connection and export of SanDisk 4GB Cruzer USB 2.0 Flash Drive. The Properties dialog box is accessed by double clicking on the device line on the main GUI interface. Specific device information is exported via the *File – Save Selected Items* or *Ctrl +S* user action. Forensic copy created with AccessData's FTK® Imager Lite (v2.9.0). Extract the SYSTEM, SOFTWARE, NTUSER.DAT, and setupapi.dev.log files. *USBDeviceForensics*© - Open program and import all of the registry files into the Open dialog box. Run program – error noted as "Unhandled exception has occurred in your application….Object reference not set to an instance of an object". Press Continue – program does not run. Test program on a fresh Windows® 7 operating system install (physical hard drive without Windows® Updates installed, isolated from the Internet). Observe same program error. Install Windows® updates including .Net framework and Service Pack 1. Program now runs. Outcome: testing to be completed on a fresh install Windows® 7. Use new and DOD wiped physical Seagate 500 GB hard drive and Macrium Reflect® (v5) imaging for OS replication purposes. Check that the program installs and is working correctly.

Date	Task	Actions, Results and General Observations
22/07/2011	Install Windows® 7 Home Premium as Test-PC on test workstation	Seagate 500 GB hard drive – Windows® 7 operating system installed and updates with no issues – 20 GB system partition only. Install updates. Disconnect Internet connection. Install Western Digital 1 TB hard drive, wiped and formatted with NTFS. Labelled as "Evidence" for all case and evidence files. Image the Seagate 500 GB hard drive with FTK® Imager Lite as a base forensic image.
05/08/2011	Conduct main tool evaluation testing of all USB 2.0 and 3.0 supported devices. Test Sequences 1 to 8	Test 1 – 1st connection of SanDisk Cruzer 4GB USB 2.0 Flash Drive at 21:13:00 hours. Run *USBDeview* and export "Live" information for device. Disconnect at 21:21:00 hours. Forensic image created, restore base image. Extract all evidence related files. MD5 checksum: 9d9406127875d582c2d9104170ce10e7
		Test 2 – 1st connect of Kingston DataTraveler 101 4GB USB flash drive at 22:00:00 hours. *USBDeview* and export "Live" information for device. Disconnect at 22:05:40 hours. Forensic image created, restore base image. Extract all evidence related files. MD5 checksum: fbc78c753fa617b47d38262ea63f2ad4
		Test 3 – 1st connection of Apacer AH325 4GB USB 2.0 Flash Drive at 22:41:29 hours. Run *USBDeview* and export "Live" information for device. Disconnect at 22:51:00 hours. Forensic image created, restore base image. Extract all evidence related files. MD5 checksum: 17a45994c441b7684d1c8eb3388db91c
		Test 4 – 1st connection of Dick Smith 2 GB USB 2.0 Micro Drive at 23:22:59 hours. Run *USBDeview* and export "Live" information for device. Disconnect at 22:26:35 hours. Forensic image created.

Date	Task	Actions, Results and General Observations
05/08/2011		Test 4 - Restore base image. Extract all evidence related files. Test 4 - MD5 checksum: 13aced40bc53a1275058463cd8979f32
06/08/2011		Test 5 - 1st connection of Transcend StoreJet 500 GB USB 2.0 PSD Device at 00:02:04 hours. *USBDeview* and export "Live" information for device. Disconnect at 00:06:00 hours. Run Forensic image created, restore base image. Extract all evidence related files. MD5 checksum: beccca0d32d5e04ca8b7c168fc290fb8
06/08/2011		Test 6 – 1st connection of Seagate 500 GB FreeAgent USB 2.0 PSD Device - Scenario 1 at 11:16:02 hours. *USBDeview* and export "Live" information for device. Disconnect at 11:22:50 hours. Forensic image created, restore base image. Extract all evidence related files. MD5 checksum: 933ef94beb48e95ebe09478b33149863
08/08/2011		Test 7 – 1st connection of 1st connection of Seagate 500 GB FreeAgent USB 2.0 PSD Device - Scenario 2. Diff date and USB 2.0 Port at 08:27:00 hours on Port_#0004. Run *USBDeview* and export "Live" information for device. Disconnect at 08:35:10. 1st connection on Port_#0003 at 08:36:15 hours. Run *USBDeview* and export "Live" information for device. Disconnect at 08:39:45 Forensic image created after last USB device disconnection from the 2nd port, MD5 checksum: 96472ac490d82f0dc5e15dfe78ba16e8 Restore base image. Extract all evidence related files.
15/08/2011		Test 8 – USB 3.0 activity only. 1st connection of Seagate 500 GB FreeAgent USB 3.0 PSD Device – Scenario 3 at 16:08:00 hours. USBDeview failed to detect this device. No information available for export.

Date	Task	Actions, Results and General Observations
15/08/2011		Restart test computer at 16:19:00 hours and reconnect device again at 16:21:09 hours. Disconnect at 16:25:59 hours. Test 8 - Forensic image created. Extract all evidence related files. MD5 checksum: 8e210167914f5ad64b7d42cc00c01467
10/09/2011	Extract and time check *USBDeview* command-line option	Complete tool evaluation data analysis by running the USBDeview command-line against the extracted SYSTEM hives for Tool Evaluation Tests 1 to 7 - USB 2.0 device Registry data. Syntax: /regfile <SYSTEM Registry File> C:\>usbdeview.exe /regfile "c:\mfit testing\test 1\registry file\system" C:\>usbdeview.exe /regfile "c:\mfit testing\test 2\registry file\system" C:\>usbdeview.exe /regfile "c:\mfit testing\test 3\registry file\system" C:\>usbdeview.exe /regfile "c:\mfit testing\test 4\registry file\system" C:\>usbdeview.exe /regfile "c:\mfit testing\test 5\registry file\system" C:\>usbdeview.exe /regfile "c:\mfit testing\test 6\registry file\system" C:\>usbdeview.exe /regfile "c:\mfit testing\test 6\registry file\system" C:\>usbdeview.exe /regfile "c:\mfit testing\test 7\registry file\system" Text files created by export function. Manual Export Timings (Stop Watch): Test 1: 01:48.66 Test 2: 01:35.62 Test 3: 01:24.78 Test 4: 01:17.54 Test 5: 00:59.87 Test 6: 01:04.45 Test 7: 01:00.22 Test 8: No data as USB 3.0 devices not supported by the software.
10/09/2011	EnCase Forensic (v6.18) manual bookmark and export of USB device artifacts	Manual Export Timings (Stop Watch): Test 1: 11:35.12 Test 2: 14:10.80 Test 3: 11:05.28 Test 4: 10:01.10 Test 5: 10:09.60

Date	Task	Actions, Results and General Observations
10/09/2011		EnCase Forensic Manual Export Timings: Test 6: 10:32.14 Test 7: 10:20.45 Test 8: 10:12.94 MD5 Hash Checks - Verified
10/09/2011	FTK RegistryViewer (v1.6.3). Normal USB Registry selections and HTML Print	Manual Timings (Stop Watch) Test 1: 03:36.63 Test 2: 03:33.61 Test 3: 03:28.86 Test 4: 03:15.17 Test 5: 02:48.10 Test 6: 02:43.83 Test 7: 02:50.34 Test 8: 02:41.58
04/11/2011	Development Field Testing of *USBDeviceForensics* prototype tool	DT – Development Test Designation. Analyst workstation date and time checked against the Industrial Research Limited (IRL) Clock Wellington. System Clock: 04/11/2011 15:30:00 IRL Clock: 04/11/2011 15:30:00 Tool timings taken from the log report of each test run. Between 1552 hours and 1635 hours connect all of the USB devices to the analyst workstation. Then run *USBForensicReporter* (prototype v1.0.6) DT001: 00:01.62 DT002: 00:01.10 DT003: 00:01.24 DT004: 00:01.31 DT005: 00:01.45 DT006: 00.01.75 DT007: – Not conducted. DT008: 00:00.66
04/11/2011	Run independent USB software protocol analyser to extract USB device descriptor information for conformation of tool evaluation	Attach each USB and PSD device to workstation with USBlyzer (v2.0) running. Capture the entire USB information for each device and export a HTML report. Of particular importance was each of the field values for: Offset Name 14 iManufacturer 15 iProduct

Date	Task	Actions, Results and General Observations
04/11/2011		USBlyzer Results (Continued) Offset Name 16 iSerialNumber A device descriptor comparison analysis confirms each evaluated tools is correctly Extracted evidence files as the USBlyzer output.
05/11/2011	Notes related to tool operation, analysis and reporting – to assist with the Gap Analysis categorisation action	*USBDeview* – Tool 1. GUI interface – easy to use with text based export function. Device descriptor information such as iSerial number was found to be accurate. The software records each port connection which was helpful for Test 7 when two different USB ports were being tested. No additional action required on the output. Cons: Only captures SYSTEM related USB device information. No reporting for SOFTWARE or NTUSER.DAT files.
		USBDeviceForensics – Tool 2. GUI interface – easy to use with CSV based export function. Only "offline" registry analysis from extracted forensic copies. Timezone setting feature is ideal for local time zone use. Device descriptor related information was found to be accurate. Windows Portable Devices sub-key of SOFTWARE hive file is not reported but the EMDMgmt sub-key last written date and time is for USB thumb drive devices. Cons: CSV export requires further formatting for reporting purposes. Dates and times that are not displayed correctly show Monday, 1 January 0001 00:00:00. Some time zone offset values are not displayed correctly – not consistent. Setupapi.dev.log entries for the test USB devices were not displayed even when each log file was selected for analysis action.

Date	Task	Actions, Results and General Observations
05/11/2011	Notes related to tool operation, analysis and reporting – to assist with the Gap Analysis categorisation action	Tool 2 (Continued) Information for PSD devices (external USB hard drives) is very limited to basic device and USBSTOR, DeviceClasses and USB sub-keys.
		EnCase® Forensic – Tool 3. Supports: EnCase® evidence file .E01 format. Local time zone settings can be set within the Case Time Zones and Time Properties options. All registry files are manually mounted via the *View File Structure* command by right-clicking on the appropriate hive file. Data can be bookmarked for reporting by sweeping the data of interest in the view pane. Allows for very comprehensive reporting of all USB evidence-related registry and system files. HTML, TXT and RTF output. The sample Enscript was very fast and automatically mounts the hive files for the tool user. Cons: Very time consuming when conducting a manual examination of the Registry and system files. The manual analysis or stock standard method of USB analysis with this tool would not be cost effective for more than five to seven USB devices. Bookmarking requires the export field values to be set and further formatting post export for reporting purposes.
		FTK®/FTK® RegistryViewer – Toolset 4.FTK® Registry Viewer allows for the majority of the registry hives to be analysed and reported on. Timezone: takes local workstation date and time setting offsets. Registry files are reported separately. Must be used as an external application with FTK v3.3 to report on *setupapi.dev.log* files. The

Date	Task	Actions, Results and General Observations
05/110/2011	Notes related to tool operation, analysis and reporting – to assist with the Gap Analysis categorisation action	Tool 3 (Continued) Cons: Reporting is confirmed to a predefined HTML report which is not easily altered for FTK RegistryViewer *FTK® 3.3* allows HTML, pdf, docx, txt and csv formats for reporting. Further formatting using Windows cut-and-past is required if *the FTK® RegistryViewer* report format is not being used as a standalone report.
06/11/2011	Compare all outputs from the toolset evaluations in August and September 2011	Print results to pdf. Each tool has a different methodology for reporting the same data. Need more consistent reporting formats and evidence set capture?
06/11/2011	Recheck forensic image copies after toolset data extractions.	All MD5 hash results match the original forensic copies.
25/11/2011 to 09/12/2011	Validation and verification testing of *USBForensicReporter* tool (v1.0.7) registry analysis and output results	Validate raw registry hive file data using WinHex, Thomassen and Norris methods. Test 1 SanDisk Cruzer 4GB USB Device data utilised along with EnCase Forensic (v6.18). Raw registry values can be manually validated but the process is very time consuming and complex. There are still some values that are not able to be deciphered even with new Carvey 2011 literature. Verification: HTML and log output from the prototype tool is consistent and accurate with the EnCase tool data output as per Appendix E.

Appendix C – Toolset Evaluation Dataset Results

TEST 1 - SanDisk 4 GB Cruzer USB 2.0 Flash Drive	
Test Details	Evidence Item: TS001 - SanDisk 4 GB Cruzer USB 2.0 Flash Drive, Serial Number: 2005304502028AB1BCA4. Connection to a Windows 7 Operating System for Evaluation and Evidence Analysis by the Sample Toolset.
Tester	Mark Simms
Test Date(s)	05 - 06 August 2011
Conditional Requirements	**CR1** - The tool supports processing of digital source evidence (i.e. evidence file or individual "Live" and "Offline" Registry Hive data).
	CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard.
	CR3 - The tool supports the examination and reporting of USB 2.0 devices
	CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool.
	CR5 - All original digital source evidence is unchanged by any subsequent tool activity or user actions.
	CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user.
	CR7 - If the tool logs processing information, the information is accurately recorded in a log file or screen output to the user.
	CR8 - The tool allows extraction of analysis and log information into a format that is viewable and usable by the user.
Source and Destination Hard Drive Information	**Source:** Seagate ST3500418AS 500 GB Hard Drive, Serial Number: 5VMCLFM3. Sectors: 976,773,168. Interface: SATA. **Destination:** Western Digital 1.0 TB WD10EALX-009BA0 Hard Drive, Serial Number: WCATR4337006. Total Sectors: 1,953,525,169. Interface: SATA
Forensic Image Hash Value	MD5 Hash Value: 9d9406127875d582c2d9104170ce10e7. Created after the Tool 1 live state action. Relevant registry and system files hashed and exported.
Post Analysis Hash Value	MD5 Hash Value: 9d9406127875d582c2d9104170ce10e7
Sample Toolset Details	Tool Name, Version, Developer Details, Usage or Licence Restrictions and Additional Software Requirements
Tool 1	USBDeview, v1.91, www.NirSoft.com, Freeware
Tool 2	USBDeviceForensics, v1.0.6, www.woanware.co.uk, Freeware
Tool 3	EnCase Forensic, v6.18, www.guidancesoftware.com, Commercial Licence, Additional EnScript: USB Device History, v0.5 - Lance Muller, Freeware
Tool 4	Registry Viewer, v1.6.3, www. accessdata.com, Commercial Licence. Used as a standalone application option within a licenced copy of Access Data's Forensic Tool Kit (FTK) v3.3 during the testing phase. A limited Demonstration Mode is downloadable from the website but has no reporting or printing functionality.
Logging and Exported Data	Data output successfully exported for all toolset examples and hash files. Refer to the individual attachments.
Tool Results	All Conditional Requirements met by Tools 3 and 4 only. No processing errors indicated by any of the toolset examples.
Test Outcomes and Comments	Tool 1 failed on CR4 as only the SYSTEM artifacts were reported - partial evidence group support only. Tool 2 also failed on CR4 as the design does not support Windows Portable Devices sub-key reporting from the SOFTWARE hive.
	Tool 2 also failed to meet all of the requirements of CR4 as the design does not support Windows Portable Devices sub-key reporting from the Software hive. Only USB information from the EMDMgmt sub-key (relating to the ReadyBoost services) is reported.
	Tool 4 only processes and reports on Registry related files. The setupapi.dev.log file is a system file that can be examined and reported on using the parent FTK 3.3 forensic software. Both tools were used in conjunction with each other.

Test 1 USBDeview SanDisk 4 GB Cruzer USB Export.txt

```
==================================================
Device Name : Port_#0004.Hub_#0005
Description : SanDisk Cruzer USB Device
Device Type : Mass Storage
Connected : Yes
Safe To Unplug : Yes
Disabled : No
USB Hub : No
Drive Letter : F:
Serial Number : 2005304502028AB1BCA4
Created Date : 5/08/2011 9:13:00 a.m.
Last Plug/Unplug Date: 5/08/2011 9:13:00 a.m.
VendorID : 0781
ProductID : 5530
Firmware Revision : 1.00
USB Class : 08
USB SubClass : 06
USB Protocol : 50
Hub / Port :
Computer Name :
Vendor Name :
Product Name :
ParentId Prefix :
Service Name : USBSTOR
Service Description: USB Mass Storage Driver
Driver Filename : USBSTOR.SYS
Device Class : USB
Device Mfg : @usbstor.inf,%generic.mfg%;Compatible USB storage
device
Power : 200 mA
Driver Description: USB Mass Storage Device
Driver Version : 6.1.7600.16385
Instance ID : USB\VID_0781&PID_5530\2005304502028AB1BCA4
==================================================
```

Test 1 USBDeviceForensics SanDisk Cruzer USB Export

Vendor: Ven_SanDisk
Product: Prod_Cruzer
Version: Rev_1.01
Serial No: 2005304502028AB1BCA4
VID: VID_0781
PID: PID_5530
ParentIdPrefix:
Drive Letter: F:
Volume Name:
GUID: ebda7496-bf42-11e0-b600-001fd0060cfd
MountPoint:
USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01#2005304502028AB1BCA4&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Install Date/Time): Friday, 5 August 2011 21:13:04
EMDMgmt Last Write Date/Time): Friday, 5 August 2011 21:13:07 (UTC+12:00) Auckland, Wellington
First Time Connected After Last Reboot (USBSTOR Date/Time): Friday, 5 August 2011 21:13:05 (UTC+12:00) Auckland, Wellington
First Time Connected After Last Reboot (DeviceClasses Date/Time): Friday, 5 August 2011 21:13:05 (UTC+12:00) Auckland, Wellington
Last Time Connected (Enum\USB VIDPID Date/Time): Friday, 5 August 2011 21:13:04 (UTC+12:00) Auckland, Wellington
Last Time Connected (MountPoints2 Date/Time): Friday, 5 August 2011 21:13:06 (UTC+12:00) Auckland, Wellington (File: NTUSER.DAT)

Test 1 - EnCase Manual Analysis

TS001 - SanDisk Cruzer 4GB Test 1 Analysis\EnCase Manual Analysis

1)
Name Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01
Last Written 05/08/2011 09:13:04p.m.
Full Path Test 1 TS001 USB Analysis\TS001\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01

```
Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01
```

2)
Name 2005304502028AB1BCA4&0
Last Written 05/08/2011 09:13:05p.m.
Full Path Test 1 TS001 USB Analysis\TS001\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0

```
2005304502028AB1BCA4&0
```

3)
Name \DosDevices\F:
Last Written
Full Path Test 1 TS001 USB Analysis\TS001\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\DosDevices\F:

```
_·?·?·_·U·S·B·S·T·O·R·#·D·i·s·k·&·V·e·n·_·S·a·n·D·i·s·k·&·P·r·o·d·_·C·r·u·z·e·r·&·R·e
·v·_·1·.·0·1·#·2·0·0·5·3·0·4·5·0·2·0·2·8·A·B·1·B·C·A·4·&·0·#·{·5·3·f·5·6·3·0·7·-·b·6·
b·f·-·1·1·d·0·-·9·4·f·2·-·0·0·a·0·c·9·1·e·f·b·8·b·}·
```

4)
Name \??\Volume{ebda7496-bf42-11e0-b600-001fd0060cfd}
Last Written
Full Path Test 1 TS001 USB Analysis\TS001\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\??\Volume{ebda7496-bf42-11e0-b600-001fd0060cfd}

```
_·?·?·_·U·S·B·S·T·O·R·#·D·i·s·k·&·V·e·n·_·S·a·n·D·i·s·k·&·P·r·o·d·_·C·r·u·z·e·r·&·R·e
·v·_·1·.·0·1·#·2·0·0·5·3·0·4·5·0·2·0·2·8·A·B·1·B·C·A·4·&·0·#·{·5·3·f·5·6·3·0·7·-·b·6·
b·f·-·1·1·d·0·-·9·4·f·2·-·0·0·a·0·c·9·1·e·f·b·8·b·}·
```

5)
Name {ebda7496-bf42-11e0-b600-001fd0060cfd}
Last Written 05/08/2011 09:13:06p.m.
Full Path Test 1 TS001 USB Analysis\TS001\Users\Test\NTUSER.DAT\NTRegistry\CMI-CreateHive{6A1C4018-979D-4291-A7DC-7AED1C75B67C}\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda7496-bf42-11e0-b600-001fd0060cfd}

```
{ebda7496-bf42-11e0-b600-001fd0060cfd}
```

Test 1 - EnCase Manual Analysis

6)
Name setupapi.dev.log
Last Written 05/08/2011 09:13:08p.m.
Full Path Test 1 TS001 USB Analysis\TS001\Windows\inf\setupapi.dev.log

```
>>>  [Device Install (Hardware initiated) - USB\VID_0781&PID_5530\2005304502028AB1BCA4]
>>>  Section start 2011/08/05 21:13:00.716
```

7)
Name ##?#USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01
#2005304502028AB1BCA4&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Last Written 05/08/2011 09:13:05p.m.
Full Path Test 1 TS001 USB Analysis\TS001\Windows\System32\config\SYSTEM\
NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91ef
b8b}\##?#USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01
#2005304502028AB1BCA4&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

```
##?#USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01#2005304502028AB1BCA4&0#{53f56307-
b6bf-11d0-94f2-00a0c91efb8b}
```

8)
Name FriendlyName
Last Written
Full Path Test 1 TS001 USB \TS001\Windows\System32\config\SOFTWARE\
NTRegistry\CMI-CreateHive{3D971F19-49AB-4000-8D39-A6D9C673D809}\Microsoft\Windo
ws Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#
VOLUME#_??_USBSTOR#DISK&VEN_SANDISK&PROD_CRUZER&REV_1.01#20053045
02028AB1BCA4&0#\FriendlyName

```
F·:·\···
```

Test 1 - EnCase Enscript Analysis

TS001 - SanDisk Cruzer 4GB Test 1 Analysis\EnCase Enscript Analysis

```
--------------------------------------------------------------------------------

The following information is from Test 1 TS001 USB Analysis\TS001\Windows\System32\config\SYSTEM:
USBSTOR:

Type   Vendor      Product     Serial_Number     Friendly_Name     USB_Driver      Last_Written_Date
ParentIDPrefix

Disk  SanDisk     Cruzer      2005304502028AB1BCA4&0  SanDisk Cruzer USB Device     05/08/2011 09:13:05p.m. NONE


\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}:

Type  Vendor      Product     Revision    Serial_Number     Driver                              Last_Written_Date

Disk  SanDisk     Cruzer      1.01  2005304502028AB1BCA4&0  {53f56307-b6bf-11d0-94f2-00a0c91efb8b}    05/08/2011
09:13:05p.m.

\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}:

Type1 Type2 Serial_Number     Signature   Offset      Length      Driver      Last_Written_Date

STORAGE     VOLUME      _??_USBSTOR DISK&VEN_SANDISK&PROD_CRUZER&REV_1.01     2005304502028AB1BCA4&0

{53F56307-B6BF-11D0-94F2-00A0C91EFB8B}    {53f5630d-b6bf-11d0-94f2-00a0c91efb8b}    05/08/2011 09:13:06p.m.


Mounted_Devices:

\DosDevices\F:    _??_USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01#2005304502028AB1BCA4&0#{53f56307-b6bf-11d0-94f2-
00a0c91efb8b}
```

Summary Report: Test 1 Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01 - FTK Registry Viewer USB Export

USBSTOR

Key Name	Name	Type	Data
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0		Key Properties	Last Written Time : 5/08/2011 9:13:05 UTC
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	Capabilities	REG_DWORD	0x00000010 (16)
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	HardwareID	REG_MULTI_SZ	USBSTOR\DiskSanDisk_Cruzer_____1.01 USBSTOR\DiskSanDisk_Cruzer_____ USBSTOR\DiskSanDisk_ USBSTOR\SanDisk_Cruzer_____1 SanDisk_Cruzer_____1 USBSTOR\GenDisk GenDisk
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk USBSTOR\RAW
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	ContainerID	REG_SZ	{f1e3bfb6-1a4c-5847-8917-e5882341356c}

ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\0008
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	Class	REG_SZ	DiskDrive
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard disk drives)
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	Service	REG_SZ	disk
ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0	FriendlyName	REG_SZ	SanDisk Cruzer USB Device

USB

Key Name	Name	Type	Data
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4		Key Properties	Last Written Time : 5/08/2011 9:13:04 UTC
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	DeviceDesc	REG_SZ	@usbstor.inf,%genericbulkonly.devicedesc%;USB Mass Storage Device
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	LocationInformation	REG_SZ	Port_#0004.Hub_#0005
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	Capabilities	REG_DWORD	0x000000D4 (212)
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	HardwareID	REG_MULTI_SZ	USB\VID_0781&PID_5530&REV_0100
			USB\VID_0781&PID_5530
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	CompatibleIDs	REG_MULTI_SZ	USB\Class_08&SubClass_06&Prot_50
			USB\Class_08&SubClass_06
			USB\Class_08
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	ContainerID	REG_SZ	{f1e3bfb6-1a4c-5847-8917-e5882341356c}
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	ConfigFlags	REG_DWORD	0x00000000 (0)

ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	ClassGUID	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	Driver	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}\0017
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	Class	REG_SZ	USB
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	Mfg	REG_SZ	@usbstor.inf,%generic.mfg%;Compatible USB storage device
ControlSet001\Enum\USB\VID_0781&PID_5530\2005304502028AB1BCA4	Service	REG_SZ	USBSTOR

DeviceClasses

Key Name	Name	Type	Data
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01#2005304502028AB1BCA4&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}		Key Properties	Last Written Time : 5/08/2011 9:13:05 UTC
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01#2005304502028AB1BCA4&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	DeviceInstance	REG_SZ	USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01\2005304502028AB1BCA4&0

MountedDevices

Key Name	Name	Type	Data
MountedDevices		Key Properties	Last Written Time : 5/08/2011 9:13:06 UTC
MountedDevices	\DosDevices\F:	REG_BINARY	5F 00 3F 00 3F 00 5F 00 55 00 53 00 42 00 53 00 54 00 4F 00 52 00 23 00 44 00 69 00 73 00 6B 00 26 00 56 00 65 00 6E 00 5F 00 53 00 61 00 6E 00 44 00 69 00 73 00 6B 00 26 00 50 00 72 00 6F 00 64 00 5F 00 43 00 72 00 75 00 7A 00 65 00 72 00 26 00 52 00 65 00 76 00 5F 00 31 00 2E 00 30 00 31 00 23 00 32 00 30 00 30 00 35 00 33 00 30 00 34 00 35 00 30 00 32 00 30 00 32 00 38 00 41 00 42 00 31 00 42 00 43 00 41 00 34 00 26 00 30 00 23 00 7B 00 35 00 33 00 66 00 35 00 36 00 33 00 30 00 37 00 2D 00 62 00 36 00 62 00 66 00 2D 00 31 00 31 00 64 00 30 00 2D 00 39 00 34 00 66 00 32 00 2D 00 30 00 30 00 61 00 30 00 63 00 39 00 31 00 65 00 66 00 62 00 38 00 62 00 7D 00
		(ASCII String)	_.?.?._.U.S.B.S.T.O.R.#.D.i.s.k.&.V.e.n._.S.a.n.D.i.s.k.&.P.r.o.d._.C.r.u.z.e.r.&.R.e.v._.1...0.1.#.2.0.0.5.3.0.4.5.0.2.0.2.8.A.B.1.B.C.A.4.&.0.#.{.5.3.f.5.6.3.0.7.-.b.6.b.f.-.1.1.d.0.-.9.4.f.2.-.0.0.a.0.c.9.1.e.f.b.8.b.}.
		(UTF-16 String)	_??_USBSTOR#Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01#2005304502028AB1BCA4&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

Summary Report: Test 1 SanDisk Cruzer - NTUSER.DAT - FTK Registry Viewer USB Export

MountPoints2

Key Name	Name	Type	Data	
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda7496-bf42-11e0-b600-001fd0060cfd}\shell		Key Properties	Last Written Time	: 5/08/2011 9:13:06 UTC
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda7496-bf42-11e0-b600-001fd0060cfd}\shell	(default)	REG_SZ	None	

Summary Report: Test 1 SanDisk Cruzer - SOFTWARE - FTK Registry Viewer USB Export

Windows Portable Devices

Key Name	Name	Type	Data
Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_?? _USBSTOR#DISK&VEN_SANDISK&PROD_CRUZER&REV_1.01#2005304502028AB1BCA4&0#		Key Properties	Last Written Time: 5/08/2011 9:13:08 UTC
Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_?? _USBSTOR#DISK&VEN_SANDISK&PROD_CRUZER&REV_1.01#2005304502028AB1BCA4&0#	FriendlyName	REG_SZ	F:\

Summary Report: Test 1 SanDisk Cruzer – Setupapi.dev.log FTK 3.3 Bookmark Extracts

Name	setupapi.dev.log
Created Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Modified Date	5/08/2011 9:13:08 p.m. (2011-08-05 09:13:08 UTC)
Accessed Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Path	TS001.E01/NONAME [NTFS]/[root]/Windows/inf/setupapi.dev.log
Exported as	files\setupapi.dev.log.txt

>>> [Device Install (Hardware initiated) - USB\VID_0781&PID_5530\2005304502028AB1BCA4]
>>> Section start 2011/08/05 21:13:00.716

Test 1 – SanDisk Cruzer USBlyzer Device Descriptor Details

USB Mass Storage Device

Connection Status Device connected
Current Configuration 1
Speed High
Device Address 1
Number Of Open Pipes 2

Device Descriptor Cruzer

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	0781h	SanDisk Corp.
10	idProduct	2	5530h	
12	bcdDevice	2	0100h	1.00
14	iManufacturer	1	01h	"SanDisk"
15	iProduct	1	02h	"Cruzer"
16	iSerialNumber	1	03h	"2005304502028AB1BCA4"
17	bNumConfigurations	1	01h	

Device Qualifier Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	0Ah	
1	bDescriptorType	1	06h	Device Qualifier
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	bNumConfigurations	1	01h	
9	bReserved	1	00h	

Configuration Descriptor 1 Bus Powered, 200 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	64h	200 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	01h	

Other Speed Configuration Descriptor 1 Bus Powered, 200 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	07h	Other Speed Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	64h	200 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

TEST 2 - Kingston 4GB DataTraveler 101 USB Device	
Test Details	Evidence Item: TS002 - Kingston DT101G2 4GB Data Traveler 101 USB 2.0 Flash Drive, Serial Number: 001A4D5F1A5CBB11200012D6. Connection to a Windows 7 Operating System for Evaluation and Evidence Analysis by the Sample Tools.
Tester	Mark Simms
Test Date(s)	05 - 06 August 2011
Conditional Requirements	**CR1** - The tool supports processing of digital source evidence (i.e. evidence file or individual "Live" and "Offline" Registry Hive data).
	CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard.
	CR3 - The tool supports the examination and reporting of USB 2.0 devices
	CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool.
	CR5 - All original digital source evidence is unchanged by any subsequent tool activity or user actions.
	CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user.
	CR7 - If the tool logs processing information, the information is accurately recorded in a log file or screen output to the user.
	CR8 - The tool allows extraction of analysis and log information into a format that is viewable and usable by the user.
Source and Destination Hard Drive Information	**Source:** Seagate ST3500418AS 500 GB Hard Drive, Serial Number: 5VMCLFM3. Sectors: 976,773,168. Interface: SATA. **Destination:** Western Digital 1.0 TB WD10EALX-009BA0 Hard Drive, Serial Number: WCATR4337006. Total Sectors: 1,953,525,169. Interface: SATA.
Forensic Image & Hash Values	MD5 Hash Value: fbc78c753fa617b47d38262e63f2ad4. Created after the Tool 1 live state action. Relevant registry and system files hashed and exported.
Post Analysis Hash Value	MD5 Hash Value: fbc78c753fa617b47d38262e63f2ad4
Sample Toolset Details	Tool Name, Version, Developer Details, Usage or Licence Restrictions and Additional Software Requirements.
Tool 1	USBDeview, v1.91, www.NirSoft.com, Freeware
Tool 2	USBDeviceForensics, v1.0.6, www.woanware.co.uk, Freeware
Tool 3	EnCase Forensic, v6.18, www.guidancesoftware.com, Commercial Licence, Additional EnScript: USB Device History, v0.5 - Lance Muller, Freeware.
Tool 4	Registry Viewer, v1.6.3, www. accessdata.com, Commercial Licence. Used as a standalone application option within a licenced copy of Access Data's Forensic Tool Kit (FTK) v3.3 during the testing phase. A limited Demonstration Mode is downloadable from the website but has no reporting or printing functionality.
Logging and Exported Data	Data output successfully exported for all toolset examples and hash files. Refer to the individual attachments.
Tool Results	All Conditional Requirements were only met by Tool 4.
Test Outcomes and Comments	Tool 1 failed on CR4 as only the SYSTEM artifacts were reported - partial evidence group support only.
	Tool 2 also failed on CR4 as the design does not support *Windows Portable Devices* sub-key reporting from the SOFTWARE hive.
	Tool 3 had 6 individual application hang-ups and crashes when moving between individual registry hives during the EnCase manual processing phase. It was very slow in processing and no errors were logged by the software - unable to meet CR6.
	Tool 4 only processes and reports on Registry related files. The setupapi.dev.log file is a system file that can be examined and reported on using the parent FTK 3.3 forensic software.

Test 2 USBDeview Kingston 4GB DataTraveler Export.txt

```
==================================================
Device Name       : Port_#0004.Hub_#0005
Description       : Kingston DT 101 G2 USB Device
Device Type       : Mass Storage
Connected         : Yes
Safe To Unplug    : Yes
Disabled          : No
USB Hub           : No
Drive Letter      : F:
Serial Number     : 001A4D5F1A5CBB11200012D6
Created Date      : 5/08/2011 10:00:00 p.m.
Last Plug/Unplug Date: 5/08/2011 10:00:02 p.m.
VendorID          : 0951
ProductID         : 1642
Firmware Revision : 1.00
USB Class         : 08
USB SubClass      : 06
USB Protocol      : 50
Hub / Port        :
Computer Name     :
Vendor Name       :
Product Name      :
ParentId Prefix   :
Service Name      : USBSTOR
Service Description: USB Mass Storage Driver
Driver Filename   : USBSTOR.SYS
Device Class      : USB
Device Mfg        : @usbstor.inf,%generic.mfg%;Compatible USB
storage device
Power             : 200 mA
Driver Description: USB Mass Storage Device
Driver Version    : 6.1.7600.16385
Instance ID       :
USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6
==================================================
```

Vendor: Ven_Kingston
Product: Prod_DT_101_G2
Version: Rev_PMAP
Serial No: 001A4D5F1A5CBB11200012D6
VID: VID_0951
PID: PID_1642
ParentIdPrefix:
Drive Letter: F:
Volume Name:
GUID: ebda74c0-bf42-11e0-b600-001fd0060cfd
MountPoint:
USBSTOR#Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP#001A4D5F1A5CBB11200012D6&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Install Date/Time): Friday, 5 August 2011 22:00:02
EMDMgmt Last Write Date/Time): Friday, 5 August 2011 10:00:04 Z (UTC)
First Time Connected After Last Reboot (USBSTOR Date/Time): Friday, 5 August 2011 10:00:03 Z (UTC)
First Time Connected After Last Reboot (DeviceClasses Date/Time): Friday, 5 August 2011 10:00:03 Z (UTC)
Last Time Connected (Enum\USB VIDPID Date/Time): Friday, 5 August 2011 10:00:02 Z (UTC)
Last Time Connected (MountPoints2 Date/Time): Friday, 5 August 2011 10:00:04 Z (UTC) (File: NTUSER.DAT)

Test 2 - EnCase Manual Analysis

TS002 - Kingston 4GB DataTraveler USB Device\EnCase Manual Analysis

1)
Name Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP
Last Written 05/08/11 22:00:02
Full Path Test 2 TS002 USB Analysis\TS002\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP

```
Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP
```

2)
Name 001A4D5F1A5CBB11200012D6&0
Last Written 05/08/11 22:00:03
Full Path Test 2 TS002 USB Analysis\TS002\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0

```
001A4D5F1A5CBB11200012D6&0
```

3)
Name \DosDevices\F:
Last Written
Full Path Test 2 TS002 USB Analysis\TS002\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\DosDevices\F:

```
_·?·?·_·U·S·B·S·T·O·R·#·D·i·s·k·&·V·e·n·_·K·i·n·g·s·t·o·n·&·P·r·o·d·_·D·T·_·1·0·1·_·G
·2·&·R·e·v·_·P·M·A·P·#·0·0·1·A·4·D·5·F·1·A·5·C·B·B·1·1·2·0·0·0·1·2·D·6·&·0·#
·{·5·3·f·5·6·3·0·7·-·b·6·b·f·-·1·1·d·0·-·9·4·f·2·-·0·0·a·0·c·9·1·e·f·b·8·b·}·
```

4)
Name \??\Volume{ebda74c0-bf42-11e0-b600-001fd0060cfd}
Last Written
Full Path Test 2 TS002 USB Analysis\TS002\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\??\Volume{ebda74c0-bf42-11e0-b600-001fd0060cfd}

```
_·?·?·_·U·S·B·S·T·O·R·#·D·i·s·k·&·V·e·n·_·K·i·n·g·s·t·o·n·&·P·r·o·d·_·D·T·_·1·0·1·_·G
·2·&·R·e·v·_·P·M·A·P·#·0·0·1·A·4·D·5·F·1·A·5·C·B·B·1·1·2·0·0·0·1·2·D·6·&·0·#·{·5·3·f·
5·6·3·0·7·-·b·6·b·f·-·1·1·d·0·-·9·4·f·2·-·0·0·a·0·c·9·1·e·f·b·8·b·}·
```

5)
Name DeviceInstance
Last Written
Full Path Test 2 TS002 USB Analysis\TS002\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP#001A4D5F1A5CBB11200012D6&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\DeviceInstance

```
U·S·B·S·T·O·R·\·D·i·s·k·&·V·e·n·_·K·i·n·g·s·t·o·n·&·P·r·o·d·_·D·T·_·1·0·1·_·G·2·&·R·e
·v·_·P·M·A·P·\·0·0·1·A·4·D·5·F·1·A·5·C·B·B·1·1·2·0·0·0·1·2·D·6·&·0··
```

Test 2 - EnCase Manual Analysis

TS002 - Kingston 4GB DataTraveler USB Device\EnCase Manual Analysis

6)
Name {ebda74c0-bf42-11e0-b600-001fd0060cfd}
Last Written 05/08/11 22:00:04
Full Path Test 2 TS002 USB Analysis\TS002\Users\Test\NTUSER.DAT\NTRegistry\CMI-CreateHive{6A1C4018-979D-4291-A7DC-7AED1C75B67C}\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda74c0-bf42-11e0-b600-001fd0060cfd}

```
{ebda74c0-bf42-11e0-b600-001fd0060cfd}
```

7)
Name setupapi.dev.log
Last Written 05/08/11 22:00:05
Full Path Test 2 TS002 USB Analysis\TS002\Windows\inf\setupapi.dev.log

```
>>>  [Device Install (Hardware initiated) -
USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6]  >>>
  Section start 2011/08/05 22:00:00.571
```

8)
Name FriendlyName
Last Written
Full Path Test 2 TS002 USB Analysis\TS002\Windows\System32\config\SOFTWARE\NTRegistry\CMI-CreateHive{3D971F19-49AB-4000-8D39-A6D9C673D809}\Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_KINGSTON&PROD_DT_101_G2&REV_PMAP#001A4D5F1A5CBB11200012D6&0#\FriendlyName

```
K•I•N•G•S•T•O•N•••
```

Test 2 - EnCase Enscript Analysis

TS002 - Kingston 4GB DataTraveler USB Device\EnCase Enscript Analysis

```
-------------------------------------------------------------------------------
The following information is from Test 2 TS002 USB Analysis\TS002\Windows\System32\config\SYSTEM:
USBSTOR:

Type   Vendor      Product     Serial_Number     Friendly_Name     USB_Driver Last_Written_Date

Disk  Kingston    DT_101_G2   001A4D5F1A5CBB11200012D6&0    Kingston DT 101 G2 USB Device 05/08/11 22:00:03

ParentIDPrefix

NONE


\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}:

Type  Vendor      Product     Revision    Serial_Number                          Driver
Disk  Kingston    DT_101_G2   PMAP  001A4D5F1A5CBB11200012D6&0    {53f56307-b6bf-11d0-94f2-00a0c91efb8b}

Last_Written_Date

05/08/11 22:00:03

\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}:

Type1 Type2 Serial_Number     Signature   Offset      Length      Driver      Last_Written_Date
STORAGE     VOLUME      _??_USBSTOR DISK&VEN_KINGSTON&PROD_DT_101_G2&REV_PMAP 001A4D5F1A5CBB11200012D6&0
      {53F56307-B6BF-11D0-94F2-00A0C91EFB8B}    {53f5630d-b6bf-11d0-94f2-00a0c91efb8b}    05/08/11 22:00:04

Mounted_Devices:

\DosDevices\F:
      _??_USBSTOR#Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP#001A4D5F1A5CBB11200012D6&0#{53f56307-b6bf-11d0-94f2-00a0
c91efb8b}
```

Summary Report: Test 2 Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP FTK Registry Viewer USB Export

USBSTOR

Key Name	Name	Type	Data
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0		Key Properties	Last Written Time : 5/08/2011 10:00:03 UTC
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	Capabilities	REG_DWORD	0x00000010 (16)
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	HardwareID	REG_MULTI_SZ	USBSTOR\DiskKingstonDT_101_G2_____PMAP USBSTOR\DiskKingstonDT_101_G2_____ USBSTOR\DiskKingston USBSTOR\KingstonDT_101_G2_____P KingstonDT_101_G2_____P USBSTOR\GenDisk GenDisk
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk USBSTOR\RAW
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	ContainerID	REG_SZ	{d5dd8676-a305-5dfa-8184-1045a7abfabf}
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\0009
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	Class	REG_SZ	DiskDrive
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard disk drives)
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	Service	REG_SZ	disk
ControlSet001\Enum\USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB11200012D6&0	FriendlyName	REG_SZ	Kingston DT 101 G2 USB Device

USB

Key Name	Name	Type	Data
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6		Key Properties	Last Written Time : 5/08/2011 10:00:02 UTC
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	DeviceDesc	REG_SZ	@usbstor.inf,%genericbulkonly.devicedesc%;USB Mass Storage Device
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	LocationInformation	REG_SZ	Port_#0004.Hub_#0005
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	Capabilities	REG_DWORD	0x000000D4 (212)
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	HardwareID	REG_MULTI_SZ	USB\VID_0951&PID_1642&REV_0100
			USB\VID_0951&PID_1642
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	CompatibleIDs	REG_MULTI_SZ	USB\Class_08&SubClass_06&Prot_50
			USB\Class_08&SubClass_06
			USB\Class_08
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	ContainerID	REG_SZ	{d5dd8676-a305-5dfa-8184-1045a7abfabf}
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	ClassGUID	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	Driver	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}\0018
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	Class	REG_SZ	USB
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	Mfg	REG_SZ	@usbstor.inf,%generic.mfg%;Compatible USB storage device
ControlSet001\Enum\USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6	Service	REG_SZ	USBSTOR

DeviceClasses

Key Name	Name	Type	Data
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP#001A4D5F1A5CBB11200012D6&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}		Key Properties	Last Written Time : 5/08/2011 10:00:03 UTC
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP#001A4D5F1A5CBB11200012D6&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	DeviceInstance	REG_SZ	USBSTOR\Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP\001A4D5F1A5CBB112

MountedDevices

Key Name	Name	Type	Data
MountedDevices		Key Properties	Last Written Time : 5/08/2011 10:00:04 UTC
MountedDevices	\DosDevices\F:	REG_BINARY	5F 00 3F 00 3F 00 5F 00 55 00 53 00 42 00 53 00 54 00 4F 00 52 00 23 00 44 00 69 00 73 00 6B 00 26 00 56 00 65 00 6E 00 5F 00 4B 00 69 00 6E 00 67 00 73 00 74 00 6F 00 6E 00 26 00 50 00 72 00 6F 00 64 00 5F 00 44 00 54 00 5F 00 31 00 30 00 31 00 5F 00 47 00 32 00 26 00 52 00 65 00 76 00 5F 00 50 00 4D 00 41 00 50 00 23 00 30 00 30 00 31 00 41 00 34 00 44 00 35 00 46 00 31 00 41 00 35 00 43 00 42 00 42 00 31 00 31 00 32 00 30 00 30 00 30 00 31 00 32 00 44 00 36 00 26 00 30 00 23 00 7B 00 35 00 33 00 66 00 35 00 36 00 33 00 30 00 37 00 2D 00 62 00 36 00 62 00 66 00 2D 00 31 00 31 00 64 00 30 00 2D 00 39 00 34 00 66 00 32 00 2D 00 30 00 30 00 61 00 30 00 63 00 39 00 31 00 65 00 66 00 62 00 38 00 62 00 7D 00
		(ASCII String)	_.?.?._.U.S.B.S.T.O.R.#.D.i.s.k.&.V.e.n._.K.i.n.g.s.t.o.n.&.P.r.o.d._.D.T._.1.0.1._.G.2.&.R.e.v._.P.M.A.P.#.0.0.1.A.4.D.5.F.1.A.5.C.B.B.1.1.2.0.0.0.1.2.D.6.&.0.#.{.5.3.f.5.6.3.0.7.-.b.6.b.f.-.1.1.d.0.-.9.4.f.2.-.0.0.a.0.c.9.1.e.f.b.8.b.}.
		(UTF-16 String)	_??_USBSTOR#Disk&Ven_Kingston&Prod_DT_101_G2&Rev_PMAP#001A4D5F1A5CBB11200012D6&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

Registry Information

Registry Viewer Report

Summary Report: Test 2 - Kingston DataTraveler - NTUSER.DAT - FTK Registry Viewer USB Export

MountPoints2

Key Name	Name	Type	Data
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda74c0-bf42-11e0-b600-001fd0060cfd}\shell		Key Properties	Last Written Time : 5/08/2011 10:00:04 UTC
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda74c0-bf42-11e0-b600-001fd0060cfd}\shell	(default)	REG_SZ	None

Summary Report: Test 2 - Kingston DataTraveler - SOFTWARE - FTK Registry Viewer USB Export

Windows Portable Devices

Key Name	Name	Type	Data
Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_KINGSTON&PROD_DT_101_G2&REV_PMAP#001A4D5F1A5CBB11200012D6&0#		Key Properties	Last Written Time: 5/08/2011 10:00:05 UTC
Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_KINGSTON&PROD_DT_101_G2&REV_PMAP#001A4D5F1A5CBB11200012D6&0#	FriendlyName	REG_SZ	KINGSTON

Summary Report: Test 2 Kingston DataTraveler – Setupapi.dev.log FTK 3.3 Bookmark Extracts

Name	setupapi.dev.log
Created Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Modified Date	5/08/2011 10:00:05 p.m. (2011-08-05 10:00:05 UTC)
Accessed Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Path	TS002.E01/NONAME [NTFS]/[root]/Windows/inf/setupapi.dev.log
Exported as	files\setupapi.dev.log.txt

>>> [Device Install (Hardware initiated) - USB\VID_0951&PID_1642\001A4D5F1A5CBB11200012D6]
>>> Section start 2011/08/05 22:00:00.571

Test 2 – Kingston DataTraveler 101 USBlyzer Device Descriptor Details

USB Mass Storage Device

Connection Status Device connected
Current Configuration 1
Speed High
Device Address 1
Number Of Open Pipes 2

Device Descriptor DT 101 G2

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	0951h	Kingston Technology
10	idProduct	2	1642h	
12	bcdDevice	2	0100h	1.00
14	iManufacturer	1	01h	"Kingston"
15	iProduct	1	02h	"DT 101 G2"
16	iSerialNumber	1	03h	"001A4D5F1A5CBB11200012D6"
17	bNumConfigurations	1	01h	

Device Qualifier Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	0Ah	
1	bDescriptorType	1	06h	Device Qualifier
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	bNumConfigurations	1	01h	
9	bReserved	1	00h	

Configuration Descriptor 1 Bus Powered, 200 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	64h	200 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Other Speed Configuration Descriptor 1 Bus Powered, 200 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	07h	Other Speed Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	64h	200 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

TEST 3 - Apacer AH325 4 GB USB 2.0 Flash Drive	
Test Details	Evidence Item: TS003 - Apacer AH3255 4 GB USB 2.0 Flash Drive, Serial Number: 000FF1103192249410006123. Connection to a Windows 7 Operating System for Evaluation and Evidence Analysis by the Sample Tools.
Tester	Mark Simms
Test Date(s)	05 - 06 August 2011
Conditional Requirements	**CR1** - The tool supports processing of digital source evidence (i.e. evidence file or individual "Live" and "Offline" Registry Hive data).
	CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard.
	CR3 - The tool supports the examination and reporting of USB 2.0 devices
	CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool.
	CR5 - All original digital source evidence is unchanged by any subsequent tool activity or user actions.
	CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user.
	CR7 - If the tool logs processing information, the information is accurately recorded in a log file or screen output to the user.
	CR8 - The tool allows extraction of analysis and log information into a format that is viewable and usable by the user.
Source and Destination Hard Drive Information	**Source:** Seagate ST3500418AS 500 GB Hard Drive, Serial Number: 5VMCLFM3. Sectors: 976,773,168. Interface: SATA. **Destination:** Western Digital 1.0 TB WD10EALX-009BA0 Hard Drive, Serial Number: WCATR4337006. Total Sectors: 1,953,525,169. Interface: SATA
Forensic Image & Hash Values	MD5 Hash Value: 17a45994c441b7684d1c8eb3388db91c. Created after the Tool 1 live state action. Relevant registry and system files hashed and exported.
Post Analysis Hash Value	MD5 Hash Value: 17a45994c441b7684d1c8eb3388db91c
Sample Toolset Details	Tool Name, Version, Developer Details, Usage or Licence Restrictions and Additional Software Requirements
Tool 1	USBDeview, v1.91, www.NirSoft.com, Freeware
Tool 2	USBDeviceForensics, v1.0.6, www.woanware.co.uk, Freeware
Tool 3	EnCase Forensic, v6.18, www.guidancesoftware.com, Commercial Licence, Additional EnScript: USB Device History, v0.5 - Lance Muller, Freeware
Tool 4	Registry Viewer, v1.6.3, www. accessdata.com, Commercial Licence. Used as a standalone application option within a licenced copy of Access Data's Forensic Tool Kit (FTK) v3.3 during the testing phase. A limited Demonstration Mode is downloadable from the website but has no reporting or printing functionality.
Logging and Exported Data	Data output successfully exported for all toolset examples and hash files. Refer to the individual attachments.
Tool Results	All Conditional Requirements met by Tools 3 and 4 only. No processing errors indicated by any of the toolset examples.
Test Outcomes and Comments	Tool 1 failed on CR4 as only the SYSTEM artifacts were reported - partial evidence group support only. Tool 2 also failed on CR4 as the design does not support *Windows Portable Devices* sub-key reporting from the SOFTWARE hive.
	Tool 4 only processes and reports on Registry related files. The setupapi.dev.log file is a system file that can be examined and reported on using the parent FTK 3.3 forensic software.

Test 3 USBDeview Apacer AH325 4 GB USB Export.txt

```
==================================================
Device Name       : Port_#0004.Hub_#0005
Description       : USB FLASH DRIVE USB Device
Device Type       : Mass Storage
Connected         : Yes
Safe To Unplug    : Yes
Disabled          : No
USB Hub           : No
Drive Letter      : F:
Serial Number     : 000FF1103192249410006123
Created Date      : 5/08/2011 10:41:29 p.m.
Last Plug/Unplug Date: 5/08/2011 10:41:31 p.m.
VendorID          : 1005
ProductID         : b113
Firmware Revision : 0.00
USB Class         : 08
USB SubClass      : 06
USB Protocol      : 50
Hub / Port        :
Computer Name     :
Vendor Name       :
Product Name      :
ParentId Prefix   :
Service Name      : USBSTOR
Service Description: USB Mass Storage Driver
Driver Filename   : USBSTOR.SYS
Device Class      : USB
Device Mfg        : @usbstor.inf,%generic.mfg%;Compatible USB
storage device
Power             : 100 mA
Driver Description: USB Mass Storage Device
Driver Version    : 6.1.7600.16385
Instance ID       :
USB\VID_1005&PID_B113\000FF1103192249410006123
==================================================
```

Vendor: Ven_USB
Product: Prod_FLASH_DRIVE
Version: Rev_1.00
Serial No: 000FF1103192249410006123
VID: VID_1005
PID: PID_B113
ParentIdPrefix:
Drive Letter: F:
Volume Name:
GUID: ebda74df-bf42-11e0-b600-001fd0060cfd
MountPoint:
USBSTOR#Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00#000FF1103192249410006123&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Install Date/Time): Friday, 5 August 2011 22:41:31
EMDMgmt Last Write Date/Time): Friday, 5 August 2011 22:41:34 (UTC+12:00) Auckland,
First Time Connected After Last Reboot (USBSTOR Date/Time): Friday, 5 August 2011 22:41:32 (UTC+12:00) Auckland, Wellington
First Time Connected After Last Reboot (DeviceClasses Date/Time): Friday, 5 August 2011 22:41:32 (UTC+12:00) Auckland, Wellington
Last Time Connected (Enum\USB VIDPID Date/Time): Friday, 5 August 2011 22:41:31 (UTC+12:00) Auckland, Wellington
Last Time Connected (MountPoints2 Date/Time): Friday, 5 August 2011 22:41:34 (UTC+12:00) Auckland, Wellington (File: NTUSER.DAT)

Test 3 - EnCase Manual Analysis

TS003 - Apacer AH325 4GB USB Device\EnCase Manual Analysis

1)
Name VID_1005&PID_B113
Last Written 05/08/11 22:41:29
Full Path Test 3 TS003 USB Analysis\TS003\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USB\VID_1005&PID_B113

VID_1005&PID_B113

2)
Name 000FF11031922494100061 23&0
Last Written 05/08/11 22:41:32
Full Path Test 3 TS003 USB Analysis\TS003\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00\000FF11031922494100061 23&0

000FF11031922494100061 23&0

3)
Name \DosDevices\F:
Last Written
Full Path Test 3 TS003 USB Analysis\TS003\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\DosDevices\F:

_·?·?·_·U·S·B·S·T·O·R·#·D·i·s·k·&·V·e·n·_·U·S·B·&·P·r·o·d·_·F·L·A·S·H·_·D·R·I·V·E·&·R·e·v·_·1·.·0·0·#·0·0·0·F·F·1·1·0·3·1·9·2·2·4·9·4·1·0·0·0·6·1·2·3·&·0·#·{·5·3·f·5·6·3·0·7·-·b·6·b·f·-·1·1·d·0·-·9·4·f·2·-·0·0·a·0·c·9·1·e·f·b·8·b·}·

4)
Name \??\Volume{ebda74df-bf42-11e0-b600-001fd0060cfd}
Last Written
Full Path Test 3 TS003 USB Analysis\TS003\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\??\Volume{ebda74df-bf42-11e0-b600-001fd0060cfd}

_·?·?·_·U·S·B·S·T·O·R·#·D·i·s·k·&·V·e·n·_·U·S·B·&·P·r·o·d·_·F·L·A·S·H·_·D·R·I·V·E·&·R·e·v·_·1·.·0·0·#·0·0·0·F·F·1·1·0·3·1·9·2·2·4·9·4·1·0·0·0·6·1·2·3·&·0·#·{·5·3·f·5·6·3·0·7·-·b·6·b·f·-·1·1·d·0·-·9·4·f·2·-·0·0·a·0·c·9·1·e·f·b·8·b·}·

5)
Name {ebda74df-bf42-11e0-b600-001fd0060cfd}
Last Written 05/08/11 22:41:34
Full Path Test 3 TS003 USB Analysis\TS003\Users\Test\NTUSER.DAT\NTRegistry\CMI-CreateHive{6A1C4018-979D-4291-A7DC-7AED1C75B67C}\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda74df-bf42-11e0-b600-001fd0060cfd}

{ebda74df-bf42-11e0-b600-001fd0060cfd}

6)
Name setupapi.dev.log
Last Written 05/08/11 22:41:36
Full Path Test 3 TS003 USB Analysis\TS003\Windows\inf\setupapi.dev.log

```
>>>  [Device Install (Hardware initiated) - USB\VID_1005&PID_B113\000FF1103192249410006123]
>>>  Section start 2011/08/05 22:41:29.684
```

Test 3 - EnCase Manual Analysis

TS003 - Apacer AH325 4GB USB Device\EnCase Manual Analysis

7)

Name	##?#USBSTOR#Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00#000FF11031922494100061 23&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Last Written	05/08/11 22:41:32
Full Path	Test 3 TS003 USB Analysis\TS003\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00#000FF11031922494100061 23&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

```
##?#USBSTOR#Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00#000FF11031922494100061 23&0#
{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
```

8)

Name FriendlyName
Last Written
Full Path Test 3 TS003 USB Analysis\TS003\Windows\System32\config\SOFTWARE\NTRegistry\CMI-CreateHive{3D971F19-49AB-4000-8D39-A6D9C673D809}\Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_USB&PROD_FLASH_DRIVE&REV_1.00#000FF11031922494100061 23&0#\FriendlyName

```
F·:·\···
```

Test 3 - EnCase Enscript Analysis

TS003 – Apacer AH325 4GB USB Device\EnCase Manual Analysis

The following information is from Test 3 TS003 USB Analysis\TS003\Windows\System32\config\SYSTEM:

USBSTOR:

Type	Vendor	Product	Serial_Number	Friendly_Name	USB_Driver	Last_Written_Date	ParentIDPrefix
Disk	USB	FLASH_DRIVE	000FF11031922494100061 23&0	USB FLASH DRIVE USB Device		05/08/11 22:41:32	NONE

\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}:

Disk	USB	FLASH_DRIVE	1.00	000FF11031922494100061 23&0	{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	05/08/11 22:41:32

\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}:

Type1	Type2	Serial_Number	Signaure	Offset	Length	Driver	Last_Written_Date
STORAGE	VOLUME	_??_USBSTOR	DISK&VEN_USB&PROD_FLASH_DRIVE&REV_1.00	000FF11031922494100061 23&0			
	{53F56307-B6BF-11D0-94F2-00A0C91EFB8B}		{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}	05/08/11 22:41:33			

Mounted_Devices:

\DosDevices\F: _??_USBSTOR#Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00#000FF11031922494100061 23&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

Summary Report: Test 3 - Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00 Apacer AH325 FTK Registry Viewer USB Export

USBSTOR

Key Name	Name	Type	Data
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00\000FF1103192249410006123&0		Key Properties	Last Written Time : 5/08/2011 10:41:32 UTC
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00\000FF1103192249410006123&0	DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00\000FF1103192249410006123&0	Capabilities	REG_DWORD	0x00000010 (16)
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00\000FF1103192249410006123&0	HardwareID	REG_MULTI_SZ	USBSTOR\DiskUSB_____FLASH_DRIVE_____1.00 USBSTOR\DiskUSB_____FLASH_DRIVE_____ USBSTOR\DiskUSB_____ USBSTOR\USB_____FLASH_DRIVE_____1 USB_____FLASH_DRIVE_____1 USBSTOR\GenDisk GenDisk
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00\000FF1103192249410006123&0	CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk USBSTOR\RAW
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00\000FF1103192249410006123&0	ContainerID	REG_SZ	{91bf8312-19ec-58d1-9d42-fe9b40fefc85}

ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00 \000FF11031922494100061 23&0	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00 \000FF11031922494100061 23&0	ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00 \000FF11031922494100061 23&0	Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\0010
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00 \000FF11031922494100061 23&0	Class	REG_SZ	DiskDrive
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00 \000FF11031922494100061 23&0	Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard disk drives)
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00 \000FF11031922494100061 23&0	Service	REG_SZ	disk
ControlSet001\Enum\USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00 \000FF11031922494100061 23&0	FriendlyName	REG_SZ	USB FLASH DRIVE USB Device

USB

Key Name	Name	Type	Data
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123		Key Properties	Last Written Time : 5/08/2011 10:41:31 UTC
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	DeviceDesc	REG_SZ	@usbstor.inf,%genericbulkonly.devicedesc%;USB Mass Storage Device
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	LocationInformation	REG_SZ	Port_#0004.Hub_#0005
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	Capabilities	REG_DWORD	0x000000D4 (212)
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	HardwareID	REG_MULTI_SZ	USB\VID_1005&PID_B113&REV_0000
			USB\VID_1005&PID_B113
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	CompatibleIDs	REG_MULTI_SZ	USB\Class_08&SubClass_06&Prot_50
			USB\Class_08&SubClass_06
			USB\Class_08
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	ContainerID	REG_SZ	{91bf8312-19ec-58d1-9d42-fe9b40fefc85}
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	ConfigFlags	REG_DWORD	0x00000000 (0)

ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	ClassGUID	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	Driver	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}\0019
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	Class	REG_SZ	USB
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	Mfg	REG_SZ	@usbstor.inf,%generic.mfg%;Compatible USB storage device
ControlSet001\Enum\USB\VID_1005&PID_B113\000FF1103192249410006123	Service	REG_SZ	USBSTOR

DeviceClasses

Key Name	Name	Type	Data
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00#000FF1103192249410006123&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}		Key Properties	Last Written Time : 5/08/2011 10:41:32 UTC
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00#000FF1103192249410006123&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	DeviceInstance	REG_SZ	USBSTOR\Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00\000FF1103192249410006123&0

MountedDevices

Key Name	Name	Type	Data
MountedDevices		Key Properties	Last Written Time : 5/08/2011 10:41:33 UTC
MountedDevices	\DosDevices\F:	REG_BINARY	5F 00 3F 00 3F 00 5F 00 55 00 53 00 42 00 53 00 54 00 4F 00 52 00 23 00 44 00 69 00 73 00 6B 00 26 00 56 00 65 00 6E 00 5F 00 55 00 53 00 42 00 26 00 50 00 72 00 6F 00 64 00 5F 00 46 00 4C 00 41 00 53 00 48 00 5F 00 44 00 52 00 49 00 56 00 45 00 26 00 52 00 65 00 76 00 5F 00 31 00 2E 00 30 00 30 00 23 00 30 00 30 00 30 00 30 00 46 00 46 00 31 00 31 00 30 00 33 00 31 00 39 00 32 00 32 00 34 00 39 00 34 00 31 00 30 00 30 00 30 00 36 00 31 00 32 00 33 00 26 00 30 00 23 00 7B 00 35 00 33 00 66 00 35 00 36 00 33 00 30 00 37 00 2D 00 62 00 36 00 62 00 66 00 2D 00 31 00 31 00 64 00 30 00 2D 00 39 00 34 00 66 00 32 00 2D 00 30 00 30 00 61 00 30 00 63 00 39 00 31 00 65 00 66 00 62 00 38 00 62 00 7D 00
		(ASCII String)	_.?.?._.U.S.B.S.T.O.R.#.D.i.s.k.&.V.e.n._.U.S.B.&.P.r.o.d._.F.L.A.S.H._.D.R.I.V.E.&.R.e.v._.1...0.0.#.0.0.0.F.F.1.1.0.3.1.9.2.2.4.9.4.1.0.0.0.6.1.2.3.&.0.#.{.5.3.f.5.6.3.0.7.-.b.6.b.f.-.1.1.d.0.-.9.4.f.2.-.0.0.a.0.c.9.1.e.f.b.8.b.}.
		(UTF-16 String)	_??_USBSTOR#Disk&Ven_USB&Prod_FLASH_DRIVE&Rev_1.00#000FF1103192249410006123&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

Summary Report: Test 3 - Apacer AH325 - NTUSER.DAT - FTK Registry Report USB Export

MountPoints2

Key Name	Name	Type	Data	
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda74df-bf42-11e0-b600-001fd0060cfd}\shell		Key Properties	Last Written Time	: 5/08/2011 10:41:34 UTC
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda74df-bf42-11e0-b600-001fd0060cfd}\shell	(default)	REG_SZ	None	

Summary Report: Test 3 - Apacer AH325 - SOFTWARE - FTK Registry Viewer USB Export

Windows Portable Devices

Key Name	Name	Type	Data
Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_?? _USBSTOR#DISK&VEN_USB&PROD_FLASH_DRIVE&REV_1.00#000FF11031922494100061 23&0#		Key Properties	Last Written Time: 5/08/2011 10:41:36 UTC
Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_?? _USBSTOR#DISK&VEN_USB&PROD_FLASH_DRIVE&REV_1.00#000FF11031922494100061 23&0#	FriendlyName	REG_SZ	F:\

Summary Report: Test 3 Apacer AH325 – Setupapi.dev.log FTK 3.3 Bookmark Extracts

Name	setupapi.dev.log
Created Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Modified Date	5/08/2011 10:41:36 p.m. (2011-08-05 10:41:36 UTC)
Accessed Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Path	TS003.E01/NONAME [NTFS]/[root]/Windows/inf/setupapi.dev.log
Exported as	files\setupapi.dev.log.txt

>>> [Device Install (Hardware initiated) - USB\VID_1005&PID_B113\000FF1103192249410006123]
>>> Section start 2011/08/05 22:41:29.684

Test 3 – Apacer AH325 USBlyzer Device Descriptor Details

USB Mass Storage Device

Connection Status Device connected
Current Configuration 1
Speed High
Device Address 1
Number Of Open Pipes 2

Device Descriptor FLASH DRIVE

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	1005h	Apacer Technology, Inc.
10	idProduct	2	B113h	
12	bcdDevice	2	0000h	0.00
14	iManufacturer	1	01h	"USB"
15	iProduct	1	02h	"FLASH DRIVE"
16	iSerialNumber	1	03h	"000FF11031922494100061 23"
17	bNumConfigurations	1	01h	

Device Qualifier Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	0Ah	
1	bDescriptorType	1	06h	Device Qualifier
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	bNumConfigurations	1	01h	
9	bReserved	1	00h	

Configuration Descriptor 1 Bus Powered, 100 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	32h	100 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Other Speed Configuration Descriptor 1 Bus Powered, 100 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	07h	Other Speed Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	32h	100 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

TEST 4 - Dick Smith 2 GB USB 2.0 Micro Drive	
Test Details	Evidence Item: TS004 - Dick Smith 2GB USB 2.0 Micro Drive, Serial Number: C7E69A7C. Connection to a Windows 7 Operating System for Evaluation and Evidence Analysis by the Sample Tools.
Tester	Mark Simms
Test Date(s)	05 - 06 August 2011
Conditional Requirements	**CR1** - The tool supports processing of digital source evidence (i.e. evidence file or individual "Live" and "Offline" Registry Hive data).
	CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard.
	CR3 - The tool supports the examination and reporting of USB 2.0 devices
	CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool.
	CR5 - All original digital source evidence is unchanged by any subsequent tool activity or user actions.
	CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user.
	CR7 - If the tool logs processing information, the information is accurately recorded in a log file or screen output to the user.
	CR8 - The tool allows extraction of analysis and log information into a format that is viewable and usable by the user.
Source and Destination Hard Drive Information	**Source:** Seagate ST3500418AS 500 GB Hard Drive, Serial Number: 5VMCLFM3. Sectors: 976,773,168. Interface: SATA. **Destination:** Western Digital 1.0 TB WD10EALX-009BA0 Hard Drive, Serial Number: WCATR4337006. Total Sectors: 1,953,525,169. Interface: SATA
Forensic Image & Hash Values	MD5 Hash Value: 13aced40bc53a1275058463cd8979f32. Created after the Tool 1 live state action. Relevant registry and system files hashed and exported.
Post Analysis Hash Value	MD5 Hash Value: 13aced40bc53a1275058463cd8979f32
Sample Toolset Details	Tool Name, Version, Developer Details, Usage or Licence Restrictions and Additional Software Requirements
Tool 1	USBDeview, v1.91, www.NirSoft.com, Freeware
Tool 2	USBDeviceForensics, v1.0.6, www.woanware.co.uk, Freeware
Tool 3	EnCase Forensic, v6.18, www.guidancesoftware.com, Commercial Licence, Additional EnScript: USB Device History, v0.5 - Lance Muller, Freeware
Tool 4	Registry Viewer, v1.6.3, www. accessdata.com, Commercial Licence. Used as a standalone application option within a licenced copy of Access Data's Forensic Tool Kit (FTK) v3.3 during the testing phase. A limited Demonstration Mode is downloadable from the website but has no reporting or printing functionality.
Logging and Exported Data	Data output successfully exported for all toolset examples and hash files. Refer to the indivdual attachments.
Tool Results	All Conditional Requirements met by Tools 3 and 4 only. No processing errors indicated by any of the toolset examples.
Test Outcomes and Comments	Tool 1 failed on CR4 as only the SYSTEM artifacts were reported - partial evidence group support only. Tool 2 also failed on CR4 as the design does not support *Windows Portable Devices* sub-key reporting from the SOFTWARE hive.
	Tool 4 only processes and reports on Registry related files. The setupapi.dev.log file is a system file that can be examined and reported on using the parent FTK 3.3 forensic software.

Test 4 USBDeview Dick Smith 2GB USB Export.txt

```
==================================================
Device Name       : Port_#0004.Hub_#0005
Description       : DS MicroDrive 2GB USB Device
Device Type       : Mass Storage
Connected         : Yes
Safe To Unplug    : Yes
Disabled          : No
USB Hub           : No
Drive Letter      : F:
Serial Number     : C7E69A7C
Created Date      : 5/08/2011 11:22:59 p.m.
Last Plug/Unplug Date: 5/08/2011 11:23:01 p.m.
VendorID          : 058f
ProductID         : 6387
Firmware Revision : 1.05
USB Class         : 08
USB SubClass      : 06
USB Protocol      : 50
Hub / Port        :
Computer Name     :
Vendor Name       :
Product Name      :
ParentId Prefix   :
Service Name      : USBSTOR
Service Description: USB Mass Storage Driver
Driver Filename   : USBSTOR.SYS
Device Class      : USB
Device Mfg        : @usbstor.inf,%generic.mfg%;Compatible USB storage
device
Power             : 100 mA
Driver Description: USB Mass Storage Device
Driver Version    : 6.1.7600.16385
Instance ID       : USB\VID_058F&PID_6387\C7E69A7C
==================================================
```

Test 4 USBDeviceForensics Dick Smith 2 GB USB Export.txt

Vendor: Ven_DS
Product: Prod_MicroDrive_2GB
Version: Rev_8.07
Serial No: C7E69A7C
VID: VID_058F
PID: PID_6387
ParentIdPrefix:
Drive Letter: F:
Volume Name:
GUID: ebda74f5-bf42-11e0-b600-001fd0060cfd
MountPoint:
USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Install Date/Time): Friday, 5 August 2011 23:23:01
EMDMgmt Last Write Date/Time): Friday, 5 August 2011 11:23:03 Z (UTC)
First Time Connected After Last Reboot (USBSTOR Date/Time): Friday, 5 August 2011 11:23:02 Z (UTC)
First Time Connected After Last Reboot (DeviceClasses Date/Time): Friday, 5 August 2011 11:23:02 Z (UTC)
Last Time Connected (Enum\USB VIDPID Date/Time): Friday, 5 August 2011 11:23:01 Z (UTC)
Last Time Connected (MountPoints2 Date/Time): Friday, 5 August 2011 11:23:03 Z (UTC)
(File: NTUSER.DAT)

Test 4 - EnCase Manual Analysis

TS004 - Dick Smith 2GB USB Device\EnCase Manual Analysis

1)
Name Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07
Last Written 05/08/11 23:23:01
Full Path Test 4 TS004 USB Analysis\TS004\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07

```
Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07
```

2)
Name C7E69A7C&0
Last Written 05/08/11 23:23:02
Full Path Test 4 TS004 USB Analysis\TS004\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0

```
C7E69A7C&0
```

3)
Name \DosDevices\F:
Last Written
Full Path Test 4 TS004 USB Analysis\TS004\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\DosDevices\F:

```
_·?·?·_·U·S·B·S·T·O·R·#·D·i·s·k·&·V·e·n·_·D·S·&·P·r·o·d·_·M·i·c·r·o·D·r·i·v·e·_·2·G·B·&·R·e·v
·_·8·.·0·7·#·C·7·E·6·9·A·7·C·&·0·#·{·5·3·f·5·6·3·0·7·-·b·6·b·f·-·1·1·d·0·-·9·4·f·2·-·0·0·a·0·
c·9·1·e·f·b·8·b·}·
```

4)
Name \??\Volume{ebda74f5-bf42-11e0-b600-001fd0060cfd}
Last Written
Full Path Test 4 TS004 USB Analysis\TS004\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\??\Volume{ebda74f5-bf42-11e0-b600-001fd0060cfd}

```
_·?·?·_·U·S·B·S·T·O·R·#·D·i·s·k·&·V·e·n·_·D·S·&·P·r·o·d·_·M·i·c·r·o·D·r·i·v·e·_·2·G·B·&·R·e·v
·_·8·.·0·7·#·C·7·E·6·9·A·7·C·&·0·#·{·5·3·f·5·6·3·0·7·-·b·6·b·f·-·1·1·d·0·-·9·4·f·2·-·0·0·a·0·
c·9·1·e·f·b·8·b·}·
```

5)
Name {ebda74f5-bf42-11e0-b600-001fd0060cfd}
Last Written 05/08/11 23:23:03
Full Path Test 4 TS004 USB Analysis\TS004\Users\Test\NTUSER.DAT\NTRegistry\CMI-CreateHive{6A1C4018-979D-4291-A7DC-7AED1C75B67C}\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda74f5-bf42-11e0-b600-001fd0060cfd}

```
{ebda74f5-bf42-11e0-b600-001fd0060cfd}
```

Test 4 - EnCase Manual Analysis

TS004 - Dick Smith 2GB USB Device\EnCase Manual Analysis

6)
Name setupapi.dev.log
Last Written 05/08/11 23:23:05
Full Path Test 4 TS004 USB Analysis\TS004\Windows\inf\setupapi.dev.log

```
>>>  [Device Install (Hardware initiated) - USB\VID_058F&PID_6387\C7E69A7C]
>>>  Section start 2011/08/05 23:22:59.179  dump: Creating Install Process: DrvInst.exe
23:22:59.179
```

7)
Name ##?#USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB &Rev_8.07#C7E69A7C&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Last Written 05/08/11 23:23:02
Full Path Test 4 TS004 USB Analysis\TS004\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

```
##?#USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0#{53f56307-b6bf-11d0-94f2-
00a0c91efb8b}
```

8)
Name DeviceInstance
Last Written
Full Path Test 4 TS004 USB Analysis\TS004\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\DeviceInstance

```
U·S·B·S·T·O·R·\·D·i·s·k·&·V·e·n·_·D·S·&·P·r·o·d·_·M·i·c·r·o·D·r·i·v·e·_·2·G·B·&·R·e·v·_·8·.·0
·7·\·C·7·E·6·9·A·7·C·&·0···
```

9)
Name FriendlyName
Last Written
Full Path Test 4 TS004 USB Analysis \TS004\Windows\System32\config\SOFTWARE\NTRegistry\CMI-CreateHive{3D971F19-49AB-4000-8D39-A6D9C673D809}\Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_DS&PROD_MICRODRIVE_2GB&REV_8.07#C7E69A7C&0#\FriendlyName

```
F·:·\···
```

Test 4 - EnCase Enscript Analysis


```
--------------------------------------------------------------------------------
The following information is from Test 4 TS004 USB Analysis\TS004\Windows\System32\config\SYSTEM:
USBSTOR:

Type   Vendor      Product     Serial_Number     Friendly_Name     USB_Driver Last_Written_Date
ParentIDPrefix

Disk  DS    MicroDrive_2GB    C7E69A7C&0  DS MicroDrive           2GB USB Device    05/08/11 23:23:02    NONE


\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}:

Type  Vendor      Product     Revision    Serial_Number     Driver      Last_Written_Date

Disk  DS    MicroDrive_2GB    8.07  C7E69A7C&0  {53f56307-b6bf-11d0-94f2-00a0c91efb8b}    05/08/11 23:23:02

\DeviceClasses\{53f5630d-b6bf-11d0-94f2-00a0c91efb8b}:

Type1 Type2 Serial_Number     Signaure    Offset      Length      Driver      Last_Written_Date

STORAGE     VOLUME      _??_USBSTOR DISK&VEN_DS&PROD_MICRODRIVE_2GB&REV_8.07  C7E69A7C&0
      {53F56307-B6BF-11D0-94F2-00A0C91EFB8B}    {53f5630d-b6bf-11d0-94f2-00a0c91efb8b}    05/08/11 23:23:03

Mounted_Devices:

\DosDevices\F:
      _??_USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
```

Summary Report: Test 4 - Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07 Dick Smith FTK Registry Viewer USB Export

USBSTOR

Key Name	Name	Type	Data
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0		Key Properties	Last Written Time : 5/08/2011 11:23:02 UTC
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	Capabilities	REG_DWORD	0x00000010 (16)
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	HardwareID	REG_MULTI_SZ	USBSTOR\DiskDS_____MicroDrive_2GB__8.07 USBSTOR\DiskDS_____MicroDrive_2GB__ USBSTOR\DiskDS_____ USBSTOR\DS_____MicroDrive_2GB__8 DS_____MicroDrive_2GB__8 USBSTOR\GenDisk GenDisk

ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk USBSTOR\RAW
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	ContainerID	REG_SZ	{d6f630b5-66fd-53dd-8c65-32f40da0c103}
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\0011
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	Class	REG_SZ	DiskDrive
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard disk drives)
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	Service	REG_SZ	disk
ControlSet001\Enum\USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07\C7E69A7C&0	FriendlyName	REG_SZ	DS MicroDrive 2GB USB Device

USB

Key Name	Name	Type	Data
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}		Key Properties	Last Written Time : 5/08/2011 11:23:02 UTC
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}			

\##? #USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0# {53f56307-b6bf-11d0-94f2-00a0c91efb8b}	DeviceInstance	REG_SZ	USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07 \C7E69A7C&0

DeviceClasses

Key Name	Name	Type	Data
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b} \##? #USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0# {53f56307-b6bf-11d0-94f2-00a0c91efb8b}		Key Properties	Last Written Time : 5/08/2011 11:23:02 UTC
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b} \##? #USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0# {53f56307-b6bf-11d0-94f2-00a0c91efb8b}	DeviceInstance	REG_SZ	USBSTOR\Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07 \C7E69A7C&0

MountedDevices

Key Name	Name	Type	Data
MountedDevices		Key Properties	Last Written Time : 5/08/2011 11:23:03 UTC
MountedDevices	\DosDevices\F:	REG_BINARY	5F 00 3F 00 3F 00 5F 00 55 00 53 00 42 00 53 00 54 00 4F 00 52 00 23 00 44 00 69 00 73 00 6B 00 26 00 56 00 65 00 6E 00 5F 00 44 00 53 00 26 00 50 00 72 00 6F 00 64 00 5F 00 4D 00 69 00 63 00 72 00 6F 00 44 00 72 00 69 00 76 00 65 00 5F 00 32 00 47 00 42 00 26 00 52 00 65 00 76 00 5F 00 38 00 2E 00 30 00 37 00 23 00 43 00 37 00 45 00 36 00 39 00 41 00 37 00 43 00 26 00 30 00 23 00 7B 00 35 00 33 00 66 00 35 00 36 00 33 00 30 00 37 00 2D 00 62 00 36 00 62 00 66 00 2D 00 31 00 31 00 64 00 30 00 2D 00 39 00 34 00 66 00 32 00 2D 00 30 00 30 00 61 00 30 00 63 00 39 00 31 00 65 00 66 00 62 00 38 00 62 00 7D 00
		(ASCII String)	_.?.?._.U.S.B.S.T.O.R.#.D.i.s.k.&.V.e.n._.D.S.&.P.r.o.d._.M.i.c.r.o.D.r.i.v.e._.2.G.B.&.R.e.v._.8...0.7.#.C.7.E.6.9.A.7.C.&.0.#.{.5.3.f.5.6.3.0.7.-.b.6.b.f.-.1.1.d.0.-.9.4.f.2.-.0.0.a.0.c.9.1.e.f.b.8.b.}.

(UTF-16 String)	_??_USBSTOR#Disk&Ven_DS&Prod_MicroDrive_2GB&Rev_8.07#C7E69A7C&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

Summary Report: Test 4 - Dick Smith - NTUSER.DAT - FTK Registry Viewer USB Export

MountPoints2

Key Name	Name	Type	Data	
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda74f5-bf42-11e0-b600-001fd0060cfd}\shell		Key Properties	Last Written Time	: 5/08/2011 11:23:03 UTC
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda74f5-bf42-11e0-b600-001fd0060cfd}\shell	(default)	REG_SZ	None	

Summary Report: Test 4 - Dick Smith - SOFTWARE - FTK Regisrty Viewer USB Export

Windows Portable Devices

Key Name	Name	Type	Data
Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_DS&PROD_MICRODRIVE_2GB&REV_8.07#C7E69A7C&0#		Key Properties	Last Written Time: 5/08/2011 11:23:05 UTC
Microsoft\Windows Portable Devices\Devices\WPDBUSENUMROOT#UMB#2&37C186B&0&STORAGE#VOLUME#_??_USBSTOR#DISK&VEN_DS&PROD_MICRODRIVE_2GB&REV_8.07#C7E69A7C&0#	FriendlyName	REG_SZ	F:\

Summary Report: Test 4 Dick Smith 2 GB – Setupapi.dev.log FTK 3.3 Bookmark Extracts

Name	setupapi.dev.log
Created Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Modified Date	5/08/2011 11:23:05 p.m. (2011-08-05 11:23:05 UTC)
Accessed Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Path	TS004.E01/NONAME [NTFS]/[root]/Windows/inf/setupapi.dev.log
Exported as	files\setupapi.dev.log.txt

>>> [Device Install (Hardware initiated) - USB\VID_058F&PID_6387\C7E69A7C]
>>> Section start 2011/08/05 23:22:59.179

Test 4 – Dick Smith USBlyzer Device Descriptor Details

USB Mass Storage Device

Connection Status Device connected
Current Configuration 1
Speed High
Device Address 1
Number Of Open Pipes 2

Device Descriptor Mass Storage

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	058Fh	Alcor Micro Corp.
10	idProduct	2	6387h	
12	bcdDevice	2	0105h	1.05
14	iManufacturer	1	01h	"Generic"
15	iProduct	1	02h	"Mass Storage"
16	iSerialNumber	1	03h	"C7E69A7C"
17	bNumConfigurations	1	01h	

Device Qualifier Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	0Ah	
1	bDescriptorType	1	06h	Device Qualifier
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	bNumConfigurations	1	01h	
9	bReserved	1	00h	

Configuration Descriptor 1 Bus Powered, 100 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	32h	100 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 01 1 Out, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	01h	1 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Endpoint Descriptor 82 2 In, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	82h	2 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Other Speed Configuration Descriptor 1 Bus Powered, 100 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	07h	Other Speed Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	32h	100 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 01 1 Out, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	01h	1 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

Endpoint Descriptor 82 2 In, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	82h	2 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

TEST 5 - Transcend StoreJet 500 GB USB 2.0 Portable Storage Device	
Test Details	Evidence Item: TS005 - Transcend StoreJet 500 USB USB 2.0 Portable Storage Device, Serial Number: 1A9306339FFF. Connection to a Windows 7 Operating System for Evaluation and Evidence Analysis by the Sample Tools.
Tester	Mark Simms
Test Date(s)	06 August 2011
Conditional Requirements	**CR1** - The tool supports processing of digital source evidence (i.e. evidence file or individual "Live" and "Offline" Registry Hive data).
	CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard.
	CR3 - The tool supports the examination and reporting of USB 2.0 devices
	CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool.
	CR5 - All original digital source evidence is unchanged by any subsequent tool activity or user actions.
	CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user.
	CR7 - If the tool logs processing information, the information is accurately recorded in a log file or screen output to the user.
	CR8 - The tool allows extraction of analysis and log information into a format that is viewable and usable by the user.
Source and Destination Hard Drive Information	**Source:** Seagate ST3500418AS 500 GB Hard Drive, Serial Number: 5VMCLFM3. Sectors: 976,773,168. Interface: SATA. **Destination:** Western Digital 1.0 TB WD10EALX-009BA0 Hard Drive, Serial Number: WCATR4337006. Total Sectors: 1,953,525,169. Interface: SATA
Forensic Image & Hash Values	MD5 Hash Value: beccca0d32d5e04ca8b7c168fc290fb8. Created after the Tool 1 live state action. Relevant registry and system files hashed and exported.
Post Analysis Hash Value	MD5 Hash Value: beccca0d32d5e04ca8b7c168fc290fb8
Sample Toolset Details	Tool Name, Version, Developer Details, Usage or Licence Restrictions and Additional Software Requirements
Tool 1	USBDeview, v1.91, www.NirSoft.com, Freeware
Tool 2	USBDeviceForensics, v1.0.6, www.woanware.co.uk, Freeware
Tool 3	EnCase Forensic, v6.18, www.guidancesoftware.com, Commercial Licence, Additional EnScript: USB Device History, v0.5 - Lance Muller, Freeware
Tool 4	Registry Viewer, v1.6.3, www. accessdata.com, Commercial Licence. Used as a standalone application option within a licenced copy of Access Data's Forensic Tool Kit (FTK) v3.3 during the testing phase. A limited Demonstration Mode is downloadable from the website but has no reporting or printing functionality.
Logging and Exported Data	Data output successfully exported for all toolset examples and hash files. Refer to the indivdual attachments.
Tool Results	All Conditional Requirements met by Tools 2, 3 and 4. No processing errors indicated by any of the toolset examples.
Test Outcomes and Comments	Tool 1 failed on CR4 as only the SYSTEM artifacts were reported - partial evidence group support only.
	Tool 4 only processes and reports on Registry related files. The setupapi.dev.log file is a system file that can be examined and reported on using the parent FTK 3.3 forensic software.
	No recent drive letter assignment entries or device serial number details were recorded in the *Window Portable Devices* sub-key of the SOFTWARE Hive file for this PSD device. The evaluation testing to date indicates only USB thumb drive entries are made. Further testing can be completed at a later time to identify other types of USB devices that are also recorded in this sub-key.

Test 5 USBDeview Transcend Jetstore 500 GB Export.txt

```
==================================================
Device Name        : Port_#0004.Hub_#0005
Description        : StoreJet Transcend USB Device
Device Type        : Mass Storage
Connected          : Yes
Safe To Unplug     : Yes
Disabled           : No
USB Hub            : No
Drive Letter       : F:
Serial Number      : 1A9306339FFF
Created Date       : 6/08/2011 12:02:04 a.m.
Last Plug/Unplug Date: 6/08/2011 12:02:06 a.m.
VendorID           : 152d
ProductID          : 2509
Firmware Revision  : 1.00
USB Class          : 08
USB SubClass       : 06
USB Protocol       : 50
Hub / Port         :
Computer Name      :
Vendor Name        :
Product Name       :
ParentId Prefix    :
Service Name       : USBSTOR
Service Description: USB Mass Storage Driver
Driver Filename    : USBSTOR.SYS
Device Class       : USB
Device Mfg         : @usbstor.inf,%generic.mfg%;Compatible USB
storage device
Power              : 2 mA
Driver Description: USB Mass Storage Device
Driver Version     : 6.1.7600.16385
Instance ID        : USB\VID_152D&PID_2509\1A9306339FFF
==================================================
```

Test 5 USBDeviceForensics Transcend StoreJet Export

Vendor: Ven_StoreJet
Product: Prod_Transcend
Version: Rev_
Serial No: 1A9306339FFF
VID: VID_152D
PID: PID_2509
ParentIdPrefix:
Drive Letter:
Volume Name:
GUID:
MountPoint:
Install Date/Time): Monday, 1 January 0001 00:00:00
EMDMgmt Last Write Date/Time): Monday, 1 January 0001 00:00:00 Z (UTC)
First Time Connected After Last Reboot (USBSTOR Date/Time): Friday, 5 August 2011 12:02:06 Z (UTC)
First Time Connected After Last Reboot (DeviceClasses Date/Time): Monday, 1 January 0001 00:00:00 Z (UTC)
Last Time Connected (Enum\USB VIDPID Date/Time): Friday, 5 August 2011 12:02:06 Z (UTC)
Last Time Connected (MountPoints2 Date/Time):

Test 5 - EnCase Manual Analysis

TS005 Transcend StoreJet 500 GB PSD\EnCase Manual Analysis

1)
Name Disk&Ven_StoreJet&Prod_Transcend&Rev_
Last Written 06/08/11 00:02:06
Full Path Test 5 TS005 USB Analysis_20110806 1604\TS005\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive {F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_

```
Disk&Ven_StoreJet&Prod_Transcend&Rev_
```

2)
Name 1A9306339FFF&0
Last Written 06/08/11 00:02:06
Full Path Test 5 TS005 USB Analysis_20110806 1604\TS005\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive {F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0

```
1A9306339FFF&0
```

3)
Name \DosDevices\F:
Last Written
Full Path Test 5 TS005 USB Analysis_20110806 1604\TS005\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive {F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\DosDevices\F:

```
k5..          (6B 35 05 0E)
```

4)
Name {ebda750c-bf42-11e0-b600-001fd0060cfd}
Last Written 06/08/11 00:02:08
Full Path Test 5 TS005 USB Analysis_20110806 1604\TS005\Users\Test\NTUSER.DAT\NTRegistry\CMI-CreateHive {6A1C4018-979D-4291-A7DC-7AED1C75B67C}\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda750c-bf42-11e0-b600-001fd0060cfd}

```
{ebda750c-bf42-11e0-b600-001fd0060cfd}
```

5)
Name setupapi.dev.log
Last Written 06/08/11 00:02:07
Full Path Test 5 TS005 USB Analysis_20110806 1604\TS005\Windows\inf\setupapi.dev.log

```
[Device Install (Hardware initiated) - USB\VID_152D&PID_2509\1A9306339FFF]
>>>  Section start 2011/08/06 00:02:04.347
```

Test 5 - EnCase Manual Analysis

6)

Name	1A9306339FFF
Last Written	06/08/11 00:02:06
Full Path	Test 5 TS005 USB Analysis_20110806 1604\TS005\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive {F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF

1A9306339FFF

Test 5 - EnCase Enscript Analysis

The following information is from Test 5 TS005 USB Analysis\TS005\Windows\System32\config\SYSTEM:
USBSTOR:

Type	Vendor	Product	Serial_Number	Friendly_Name	USB_Driver	Last_Written_Date	ParentIDPrefix
Disk	StoreJet	Transcend	1A9306339FFF&0	StoreJet Transcend USB Device		06/08/11 00:02:06	NONE

\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}:

Type	Vendor	Product	Serial_Number	Driver	Last_Written_Date
Disk	StoreJet	Transcend	1A9306339FFF&0	{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	06/08/11 00:02:06

Mounted_Devices:

\DosDevices\F: DiskSignature: e05356b VolumeByteOffsetStart: 32256

Summary Report: Test 5 - Disk&Ven_StoreJet&Prod_Transcend&Rev_ FTK Registry Viewer USB Export

USBSTOR

Key Name	Name	Type	Data
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0		Key Properties	Last Written Time : 5/08/2011 12:02:06 UTC
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	Capabilities	REG_DWORD	0x00000010 (16)
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	HardwareID	REG_MULTI_SZ	USBSTOR\DiskStoreJetTranscend_____
			USBSTOR\DiskStoreJetTranscend_____
			USBSTOR\DiskStoreJet
			USBSTOR\StoreJetTranscend_____
			StoreJetTranscend_____
			USBSTOR\GenDisk
			GenDisk
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk
			USBSTOR\RAW

ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	ContainerID	REG_SZ	{5bf8b9e8-bbd6-5cbc-bad9-3a3a8d387041}
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\0012
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	Class	REG_SZ	DiskDrive
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard disk drives)
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	Service	REG_SZ	disk
ControlSet001 \Enum\USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0	FriendlyName	REG_SZ	StoreJet Transcend USB Device

USB

Key Name	Name	Type	Data
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF		Key Properties	Last Written Time : 5/08/2011 12:02:06 UTC
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	DeviceDesc	REG_SZ	@usbstor.inf,%genericbulkonly.devicedesc%;USB Mass Storage Device
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	LocationInformation	REG_SZ	Port_#0004.Hub_#0005
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	Capabilities	REG_DWORD	0x000000D4 (212)
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	HardwareID	REG_MULTI_SZ	USB\VID_152D&PID_2509&REV_0100 USB\VID_152D&PID_2509
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	CompatibleIDs	REG_MULTI_SZ	USB\Class_08&SubClass_06&Prot_50

			USB\Class_08&SubClass_06
			USB\Class_08
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	ContainerID	REG_SZ	{5bf8b9e8-bbd6-5cbc-bad9-3a3a8d387041}
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	ClassGUID	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	Driver	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}\0021
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	Class	REG_SZ	USB
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	Mfg	REG_SZ	@usbstor.inf,%generic.mfg%;Compatible USB storage device
ControlSet001\Enum\USB\VID_152D&PID_2509\1A9306339FFF	Service	REG_SZ	USBSTOR

DeviceClasses

Key Name	Name	Type	Data
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_StoreJet&Prod_Transcend&Rev_#1A9306339FFF&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}		Key Properties	Last Written Time : 5/08/2011 12:02:06 UTC
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_StoreJet&Prod_Transcend&Rev_#1A9306339FFF&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	DeviceInstance	REG_SZ	USBSTOR\Disk&Ven_StoreJet&Prod_Transcend&Rev_\1A9306339FFF&0

MountedDevices

Key Name	Name	Type	Data
MountedDevices		Key Properties	Last Written Time : 5/08/2011 12:02:07 UTC

MountedDevices	\DosDevices\F:	REG_BINARY	6B 35 05 0E 00 7E 00 00 00 00 00 00
		(ASCII String)	k5...~......
		(UTF-16 String)	㕫ค縀

Summary Report: Test 5 - Transcend StoreJet - NTUSER.DAT - FTK Registy Viewer USB Export

MountPoints2

Key Name	Name	Type	Data	
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda750c-bf42-11e0-b600-001fd0060cfd}\shell		Key Properties	Last Written Time	: 5/08/2011 12:02:08 UTC
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{ebda750c-bf42-11e0-b600-001fd0060cfd}\shell	(default)	REG_SZ	None	

Summary Report: Test 5 - Transcend StoreJet SanDisk Cruzer – Setupapi.dev.log FTK 3.3 Bookmark Extracts

Name	setupapi.dev.log
Created Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Modified Date	6/08/2011 12:02:07 a.m. (2011-08-05 12:02:07 UTC)
Accessed Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Path	TS005.E01/NONAME [NTFS]/[root]/Windows/inf/setupapi.dev.log
Exported as	files\setupapi.dev.log.txt

>>> [Device Install (Hardware initiated) - USB\VID_152D&PID_2509\1A9306339FFF]
>>> Section start 2011/08/06 00:02:04.347

Test 5 – Transcend StoreJet USBlyzer Device Descriptor Details

USB Mass Storage Device

Connection Status Device connected
Current Configuration 1
Speed High
Device Address 1
Number Of Open Pipes 2

Device Descriptor StoreJet Transcend

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	152Dh	JMicron Technology Corp. / JMicron USA Technology Corp.
10	idProduct	2	2509h	
12	bcdDevice	2	0100h	1.00
14	iManufacturer	1	0Ah	"StoreJet Transcend"
15	iProduct	1	0Bh	"StoreJet Transcend"
16	iSerialNumber	1	05h	"1A9306339FFF"
17	bNumConfigurations	1	01h	

Device Qualifier Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	0Ah	
1	bDescriptorType	1	06h	Device Qualifier
2	bcdUSB	2	0200h	USB Spec 2.0
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	bNumConfigurations	1	01h	
9	bReserved	1	00h	

Configuration Descriptor 1 Self Powered

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	04h	"USB Mass Storage"
7	bmAttributes	1	C0h	Self Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.1......	Yes
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	01h	2 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	06h	"MSC Bulk-Only Transfer"

Endpoint Descriptor 81 1 In, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Other Speed Configuration Descriptor 1 Self Powered

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	07h	Other Speed Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	04h	"USB Mass Storage"
7	bmAttributes	1	C0h	Self Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.1......	Yes
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	01h	2 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	06h	"MSC Bulk-Only Transfer"

Endpoint Descriptor 81 1 In, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 64 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0040h	64 bytes
6	bInterval	1	00h	

TEST 6 - Seagate 500 GB FreeAgent GoFlex USB 2.0/3.0 Portable Storage Device	
Test Details	Evidence Item: TS006 - Seagate 500 GB FreeAgent GoFlex USB 2.0/3.0 Portable Storage Device, Serial Number: NA0ENSYX. Connection to a Windows 7 Operating System for Evaluation and Evidence Analysis by the Sample Tools.
Tester	Mark Simms
Test Date(s)	06 August 2011
Conditional Requirements	**CR1** - The tool supports processing of digital source evidence (i.e. evidence file or individual "Live" and "Offline" Registry Hive data).
	CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard.
	CR3 - The tool supports the examination and reporting of USB 2.0 devices
	CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool.
	CR5 - All original digital source evidence is unchanged by any subsequent tool activity or user actions.
	CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user.
	CR7 - If the tool logs processing information, the information is accurately recorded in a log file or screen output to the user.
	CR8 - The tool allows extraction of analysis and log information into a format that is viewable and usable by the user.
Source and Destination Hard Drive Information	**Source:** Seagate ST3500418AS 500 GB Hard Drive, Serial Number: 5VMCLFM3. Sectors: 976,773,168. Interface: SATA. **Destination:** Western Digital 1.0 TB WD10EALX-009BA0 Hard Drive, Serial Number: WCATR4337006. Total Sectors: 1,953,525,169. Interface: SATA
Forensic Image & Hash Values	MD5 Hash Value: 933ef94beb48e95ebe09478b33149863. Created after the Tool 1 live state action. Relevant registry and system files hashed and exported.
Post Analysis Hash Value	MD5 Hash Value: 933ef94beb48e95ebe09478b33149863
Sample Toolset Details	Tool Name, Version, Developer Details, Usage or Licence Restrictions and Additional Software Requirements
Tool 1	USBDeview, v1.91, www.NirSoft.com, Freeware
Tool 2	USBDeviceForensics, v1.0.6, www.woanware.co.uk, Freeware
Tool 3	EnCase Forensic, v6.18, www.guidancesoftware.com, Commercial Licence, Additional EnScript: USB Device History, v0.5 - Lance Muller, Freeware
Tool 4	Registry Viewer, v1.6.3, www. accessdata.com, Commercial Licence. Used as a standalone application option within a licenced copy of Access Data's Forensic Tool Kit (FTK) v3.3 during the testing phase. A limited demonstration application is downloadable from the website but has no reporting or printing functionality.
Logging and Exported Data	Data output successfully exported for all toolset examples and hash files. Refer to the indivdual attachments.
Tool Results	All Conditional Requirements met by Tools 2, 3 and 4. No processing errors indicated by any of the toolset examples. No PSD *Windows Portable Devices* capture in the SOFTWARE hive file.
Test Outcomes and Comments	USB 2.0 PSD Testing Only - Scenario 1
	Tool 1 again failed on CR4 as only the SYSTEM artifacts were reported - partial evidence group support only.
	Tool 4 only processes and reports on Registry related files. The setupapi.dev.log file is a system file that can be examined and reported on using the parent FTK 3.3 forensic software.

Test 6 USBDeview Seagate FreeAgent GoFlex Export.txt

```
==================================================
Device Name       : Port_#0004.Hub_#0005
Description       : Seagate FreeAgent GoFlex USB Device
Device Type       : Mass Storage
Connected         : Yes
Safe To Unplug    : Yes
Disabled          : No
USB Hub           : No
Drive Letter      : F:
Serial Number     : NA0ENSYX
Created Date      : 6/08/2011 11:16:05 a.m.
Last Plug/Unplug Date: 6/08/2011 11:16:09 a.m.
VendorID          : 0bc2
ProductID         : 5031
Firmware Revision : 1.00
USB Class         : 08
USB SubClass      : 06
USB Protocol      : 50
Hub / Port        :
Computer Name     :
Vendor Name       :
Product Name      :
ParentId Prefix   :
Service Name      : USBSTOR
Service Description: USB Mass Storage Driver
Driver Filename   : USBSTOR.SYS
Device Class      : USB
Device Mfg        : @usbstor.inf,%generic.mfg%;Compatible USB
storage device
Power             : 500 mA
Driver Description: USB Mass Storage Device
Driver Version    : 6.1.7600.16385
Instance ID       : USB\VID_0BC2&PID_5031\NA0ENSYX
==================================================
```

Test 6 USBDeviceForensics Seagate FreeAgent GoFlex PSD Export

---
Vendor: Ven_Seagate
Product: Prod_FreeAgent_GoFlex
Version: Rev__210
Serial No: NA0ENSYX
VID: VID_0BC2
PID: PID_5031
ParentIdPrefix:
Drive Letter:
Volume Name:
GUID:
MountPoint:
Install Date/Time): Monday, 1 January 0001 00:00:00
EMDMgmt Last Write Date/Time): Monday, 1 January 0001 00:00:00 Z (UTC)
First Time Connected After Last Reboot (USBSTOR Date/Time): Friday, 5 August 2011 23:16:10 Z (UTC)
First Time Connected After Last Reboot (DeviceClasses Date/Time): Monday, 1 January 0001 00:00:00 Z (UTC)
Last Time Connected (Enum\USB VIDPID Date/Time): Friday, 5 August 2011 23:16:09 Z (UTC)

Last Time Connected (MountPoints2 Date/Time):

Test 6 - EnCase Manual Analysis

TS006 - Seagate GoFlex 500 GB USB 2.0/3.0 PSD Device\EnCase Manual Analysis

1)
Name Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210
Last Written 06/08/11 11:16:09
Full Path Test 6 TS006 USB Analysis\TS006\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210

```
Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210
```

2)
Name NA0ENSYX&0
Last Written 06/08/11 11:16:10
Full Path Test 6 TS006 USB Analysis\TS006\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0

```
NA0ENSYX&0
```

3)
Name \DosDevices\F:
Last Written
Full Path Test 6 TS006 USB Analysis\TS006\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\DosDevices\F:

```
30 05 54 C5
```

4)
Name \??\Volume{a2c76902-bfb8-11e0-825b-001fd0060cfd}
Last Written
Full Path Test 6 TS006 USB Analysis\TS006\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\??\Volume{a2c76902-bfb8-11e0-825b-001fd0060cfd}

```
30 05 54 C5
```

5)
Name {a2c76902-bfb8-11e0-825b-001fd0060cfd}
Last Written 06/08/11 11:16:12
Full Path Test 6 TS006 USB Analysis\TS006\Users\Test\NTUSER.DAT\NTRegistry\CMI-CreateHive{6A1C4018-979D-4291-A7DC-7AED1C75B67C}\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{a2c76902-bfb8-11e0-825b-001fd0060cfd}

```
{a2c76902-bfb8-11e0-825b-001fd0060cfd}
```

6)
Name setupapi.dev.log
Last Written 06/08/11 11:20:45
Full Path Test 6 TS006 USB Analysis\TS006\Windows\inf\setupapi.dev.log

```
 [Device Install (Hardware initiated) - USB\VID_0BC2&PID_5031\NA0ENSYX]
>>>  Section start 2011/08/06 11:16:06.001
```

Test 6 - EnCase Manual Analysis

TS006 - Seagate GoFlex 500 GB USB 2.0/3.0 PSD Device\EnCase Manual Analysis

8)

Name	##?#USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex &Rev__210#NA0ENSYX&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}
Last Written	06/08/11 11:16:10
Full Path	Test 6 TS006 USB Analysis\TS006\Windows\System32\config\SYSTEM\ NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91ef b8b}\##?#USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex& Rev__210#NA0ENSYX&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

```
##?#USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210#NA0ENSYX&0#{53f56307-
b6bf-11d0-94f2-00a0c91efb8b}
```

Test 6 - EnCase Enscript Analysis

TS006 - Seagate GoFlex 500 GB USB 2.0/3.0 PSD Device\EnCase Enscript Analysis

```
-----------------------------------------------------------------------------------------------------
The following information is from Test 6 TS006 USB Analysis\TS006\Windows\System32\config\SYSTEM:
USBSTOR:

Type   Vendor      Product     Serial_Number     Friendly_Name     USB_Driver      Last_Written_Date  ParentIDPrefix

Disk  Seagate     FreeAgent_GoFlex  NA0ENSYX&0  Seagate FreeAgent  GoFlex USB Device      06/08/11 11:16:10 NONE


\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}:

Type  Vendor      Product     Revision    Serial_Number           Driver                          Last_Written_Date

Disk  Seagate     FreeAgent_GoFlex  _210  NA0ENSYX&0  {53f56307-b6bf-11d0-94f2-00a0c91efb8b}    06/08/11 11:16:10


Mounted_Devices:

\DosDevices\F:    DiskSignature: c5540530 VolumeByteOffsetStart: 32256
```

Summary Report: Test 6 - Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210 - FTK Registry Viewer PSD Export

USBSTOR

Key Name	Name	Type	Data
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0		Key Properties	Last Written Time : 5/08/2011 23:16:10 UTC
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Capabilities	REG_DWORD	0x00000010 (16)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	HardwareID	REG_MULTI_SZ	USBSTOR\DiskSeagate_FreeAgent_GoFlex_210 USBSTOR\DiskSeagate_FreeAgent_GoFlex USBSTOR\DiskSeagate_ USBSTOR\Seagate_FreeAgent_GoFlex_ Seagate_FreeAgent_GoFlex_ USBSTOR\GenDisk GenDisk
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk USBSTOR\RAW

ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	ContainerID	REG_SZ	{2b43eb40-fe0e-5248-97bb-41455ca30041}
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\0013
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Class	REG_SZ	DiskDrive
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard disk drives)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Service	REG_SZ	disk
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	FriendlyName	REG_SZ	Seagate FreeAgent GoFlex USB Device

USB

Key Name	Name	Type	Data
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX		Key Properties	Last Written Time : 5/08/2011 23:16:09 UTC
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	DeviceDesc	REG_SZ	@usbstor.inf,%genericbulkonly.devicedesc%;USB Mass Storage Device
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	LocationInformation	REG_SZ	Port_#0004.Hub_#0005
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Capabilities	REG_DWORD	0x000000D4 (212)
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	HardwareID	REG_MULTI_SZ	USB\VID_0BC2&PID_5031&REV_0100 USB\VID_0BC2&PID_5031
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	CompatibleIDs	REG_MULTI_SZ	USB\Class_08&SubClass_06&Prot_50 USB\Class_08&SubClass_06

			USB\Class_08
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	ContainerID	REG_SZ	{2b43eb40-fe0e-5248-97bb-41455ca30041}
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	ClassGUID	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Driver	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}\0022
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Class	REG_SZ	USB
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Mfg	REG_SZ	@usbstor.inf,%generic.mfg%;Compatible USB storage device
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Service	REG_SZ	USBSTOR

DeviceClasses

Key Name	Name	Type	Data
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_StoreJet&Prod_Transcend&Rev_#1A9306339FFF&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\#		Key Properties	Last Written Time : 5/08/2011 12:02:06 UTC
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_StoreJet&Prod_Transcend&Rev_#1A9306339FFF&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\#	SymbolicLink	REG_SZ	\\?\USBSTOR#Disk&Ven_StoreJet&Prod_Transcend&Rev_#1A9306339FFF&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}

MountedDevices

Key Name	Name	Type	Data
MountedDevices		Key Properties	Last Written Time : 5/08/2011 23:16:11 UTC
MountedDevices	\DosDevices\F:	REG_BINARY	30 05 54 C5 00 7E 00 00 00 00 00 00

(ASCII String) 0.T..~.....

(UTF-16 String) □암緞

Summary Report: Test 6 - Seagate GoFlex - NTUSER.DAT - FTK Registry Viewer PSD Export

MountPoints2

Key Name	Name	Type	Data	
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{a2c76902-bfb8-11e0-825b-001fd0060cfd}\shell		Key Properties	Last Written Time	: 5/08/2011 23:16:12 UTC
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{a2c76902-bfb8-11e0-825b-001fd0060cfd}\shell	(default)	REG_SZ	None	

Summary Report: Test 6 - Seagate GoFlex – Setupapi.dev.log FTK 3.3 Bookmark Extracts

Name	setupapi.dev.log
Created Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Modified Date	6/08/2011 11:20:45 a.m. (2011-08-05 23:20:45 UTC)
Accessed Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Path	TS006.E01/NONAME [NTFS]/[root]/Windows/inf/setupapi.dev.log
Exported as	files\setupapi.dev.log.txt

>>> [Device Install (Hardware initiated) - USB\VID_0BC2&PID_5031\NA0ENSYX]
>>> Section start 2011/08/06 11:16:06.001

Test 6 – Seagate GoFlex USBlyzer Device Descriptor Details

USB Mass Storage Device

Connection Status Device connected
Current Configuration 1
Speed High
Device Address 1
Number Of Open Pipes 2

Device Descriptor FreeAgent GoFlex

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0210h	Invalid revision
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	0BC2h	Seagate RSS LLC
10	idProduct	2	5031h	
12	bcdDevice	2	0100h	1.00
14	iManufacturer	1	02h	"Seagate"
15	iProduct	1	03h	"FreeAgent GoFlex"
16	iSerialNumber	1	01h	"NA0ENSYX"
17	bNumConfigurations	1	01h	

Configuration Descriptor 1 Bus Powered, 500 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	FAh	500 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

TEST 7 - Seagate 500 GB FreeAgent GoFlex USB 2.0/3.0 Portable Storage Device	
Test Details	Evidence Item: TS007 - Seagate 500 GB FreeAgent GoFlex USB 2.0/3.0 Portable Storage Device, Serial Number: NA0ENSYX. Continuation from Test 6 using the same USB Device - Scenario 2 - USB 2.0 Only. Part 1 - Connection to a Windows 7 Operating System for Evaluation and Evidence Analysis by the Sample Tools on a different date. Part 2 - Connection to a Windows 7 Operating System using a Different USB Port.
Tester	Mark Simms
Test Date(s)	08 August 2011
Conditional Requirements	**CR1** - The tool supports processing of digital source evidence (i.e. evidence file or individual "Live" and "Offline" Registry Hive data).
	CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard.
	CR3 - The tool supports the examination and reporting of USB 2.0 devices
	CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool.
	CR5 - All original digital source evidence is unchanged by any subsequent tool activity or user actions.
	CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user.
	CR7 - If the tool logs processing information, the information is accurately recorded in a log file or screen output to the user.
	CR8 - The tool allows extraction of analysis and log information into a format that is viewable and usable by the user.
Source and Destination Hard Drive Information	**Source:** Seagate ST3500418AS 500 GB Hard Drive, Serial Number: 5VMCLFM3. Sectors: 976,773,168. Interface: SATA. **Destination:** Western Digital 1.0 TB WD10EALX-009BA0 Hard Drive, Serial Number: WCATR4337006. Total Sectors: 1,953,525,169. Interface: SATA
Forensic Image & Hash Values	MD5 Hash Value: 96472ac490d82f0dc5e15dfe78ba16e8. Created after the Tool 1 live state action. Relevant registry and system files hashed and exported.
Post Analysis Hash Value	MD5 Hash Value: 96472ac490d82f0dc5e15dfe78ba16e8
Sample Toolset Details	Tool Name, Version, Developer Details, Usage or Licence Restrictions and Additional Software Requirements
Tool 1	USBDeview, v1.91, www.NirSoft.com, Freeware
Tool 2	USBDeviceForensics, v1.0.6, www.woanware.co.uk, Freeware
Tool 3	EnCase Forensic, v6.18, www.guidancesoftware.com, Commercial Licence, Additional EnScript: USB Device History, v 0.5 - Lance Muller, Freeware
Tool 4	FTK RegistryViewer, v1.6.3, www. accessdata.com, Commercial Licence. Used as a standalone application option within a licenced copy of Access Data's Forensic Tool Kit (FTK) v3.3 during the testing phase. A limited demonstration application is downloadable from the website but has no reporting or printing functionality.
Logging and Exported Data	Data output successfully exported for all toolset examples and hash files. Refer to the individual attachments.
Tool Results	All Conditional Requirements met by Tools 3 and 4. No processing errors indicated by any of the toolset examples.

TEST 7 - Seagate 500 GB FreeAgent GoFlex USB 2.0/3.0 Portable Storage Device (Continued)	
Test Outcomes and Comments	Tool 1 again failed on CR4 as only the SYSTEM artifacts were reported - partial evidence group support only.
	Tool 4 only processes and reports on Registry related files. The setupapi.dev.log file is a system file that can be examined and reported on using the parent FTK 3.3 forensic software.
	No PSD *Windows Portable Devices capture* in the SOFTWARE hive file for this device. Comparison Analysis Results for Each Scenario: 06/08/2011 - Test 6 - Scenario 1 - 1st PSD Connection: ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX\ LocationInformation = Port_#0004.Hub_#0005 08/08/2011 - Test 7 - Scenario 2 - 2nd PSD Connection on another date from the 1st connection: ControlSet001\Enum\USB\VID_0BC2&PID_5031\ NA0ENSYX\LocationInformation = Port_#0003.Hub_#0005 Results: The only difference when connecting the same device across two different dates and ports is the **locationInformation** values in the USB sub-key are updated to reflect the new port location. No other USB or USBSTOR data values are changed apart from the respective Last Written date and time stamps.

Test 7 USBDeview Seagate FreeAgent GoFlex PSD Export Part 1 Port 1.txt

```
==================================================
Device Name       : Port_#0004.Hub_#0005
Description       : Seagate FreeAgent GoFlex USB Device
Device Type       : Mass Storage
Connected         : Yes
Safe To Unplug    : Yes
Disabled          : No
USB Hub           : No
Drive Letter      : F:
Serial Number     : NA0ENSYX
Created Date      : 6/08/2011 11:16:05 a.m.
Last Plug/Unplug Date: 8/08/2011 8:27:05 a.m.
VendorID          : 0bc2
ProductID         : 5031
Firmware Revision : 1.00
USB Class         : 08
USB SubClass      : 06
USB Protocol      : 50
Hub / Port        :
Computer Name     :
Vendor Name       :
Product Name      :
ParentId Prefix   :
Service Name      : USBSTOR
Service Description: USB Mass Storage Driver
Driver Filename   : USBSTOR.SYS
Device Class      : USB
Device Mfg        : @usbstor.inf,%generic.mfg%;Compatible USB
storage device
Power             : 500 mA
Driver Description: USB Mass Storage Device
Driver Version    : 6.1.7600.16385
Instance ID       : USB\VID_0BC2&PID_5031\NA0ENSYX
==================================================
```

Test 7 USBDeview Seagate FreeAgent GoFlex PSD Export Part 2 Port 2.txt

```
==================================================
Device Name       : Port_#0003.Hub_#0005
Description       : Seagate FreeAgent GoFlex USB Device
Device Type       : Mass Storage
Connected         : Yes
Safe To Unplug    : Yes
Disabled          : No
USB Hub           : No
Drive Letter      : F:
Serial Number     : NA0ENSYX
Created Date      : 6/08/2011 11:16:05 a.m.
Last Plug/Unplug Date: 8/08/2011 8:36:20 a.m.
VendorID          : 0bc2
ProductID         : 5031
Firmware Revision : 1.00
USB Class         : 08
USB SubClass      : 06
USB Protocol      : 50
Hub / Port        :
Computer Name     :
Vendor Name       :
Product Name      :
ParentId Prefix   :
Service Name      : USBSTOR
Service Description: USB Mass Storage Driver
Driver Filename   : USBSTOR.SYS
Device Class      : USB
Device Mfg        : @usbstor.inf,%generic.mfg%;Compatible USB
storage device
Power             : 500 mA
Driver Description: USB Mass Storage Device
Driver Version    : 6.1.7600.16385
Instance ID       : USB\VID_0BC2&PID_5031\NA0ENSYX
==================================================
```

Test 7 USBDeviceForensics Seagate FreeAgent GoFlex PSD Export

---
Vendor: Ven_Seagate
Product: Prod_FreeAgent_GoFlex
Version: Rev__210
Serial No: NA0ENSYX
VID: VID_0BC2
PID: PID_5031
ParentIdPrefix:
Drive Letter:
Volume Name:
GUID:
MountPoint:
Install Date/Time): Monday, 1 January 0001 00:00:00
EMDMgmt Last Write Date/Time): Monday, 1 January 0001 00:00:00 Z (UTC)
First Time Connected After Last Reboot (USBSTOR Date/Time): Sunday, 7 August 2011 20:27:05 Z
First Time Connected After Last Reboot (DeviceClasses Date/Time): Monday, 1 January 0001 00:0
Last Time Connected (Enum\USB VIDPID Date/Time): Sunday, 7 August 2011 20:36:20 Z (UTC)
Last Time Connected (MountPoints2 Date/Time):

Test 7 - EnCase Manual Analysis

TS007 – Seagate GoFlex 500 GB USB 2.0/3.0 PSD Device\EnCase Manual Analysis

1)
Name Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210
Last Written 06/08/11 11:16:09
Full Path Test 7 TS007 USB Analysis\TS007\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive {F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210

```
Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210
```

2)
Name NA0ENSYX&0
Last Written 08/08/11 08:27:05
Full Path Test 7 TS007 USB Analysis\TS007\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive {F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0

```
NA0ENSYX&0
```

3)
Name \DosDevices\F:
Last Written
Full Path Test 7 TS007 USB Analysis\TS007\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive {F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\DosDevices\F:

```
30 05 54 C5
```

4)
Name \??\Volume{a2c76902-bfb8-11e0-825b-001fd0060cfd}
Last Written
Full Path Test 7 TS007 USB Analysis\TS007\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive {F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\??\Volume{a2c76902-bfb8-11e0-825b-001fd0060cfd}

```
30 05 54 C5
```

5)
Name {a2c76902-bfb8-11e0-825b-001fd0060cfd}
Last Written 08/08/11 08:36:21
Full Path Test 7 TS007 USB Analysis\TS007\Users\Test\NTUSER.DAT\NTRegistry\CMI-CreateHive {6A1C4018-979D-4291-A7DC-7AED1C75B67C}\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{a2c76902-bfb8-11e0-825b-001fd0060cfd}

```
{a2c76902-bfb8-11e0-825b-001fd0060cfd}
```

Test 7 - EnCase Manual Analysis

6)
Name setupapi.dev.log
Last Written 08/08/11 09:00:32
Full Path Test 7 TS007 USB Analysis_20110814 1559\TS007\Windows\inf\setupapi.dev.log

```
[Boot Session: 2011/08/06 11:14:12.359]

>>>  [Device Install (Hardware initiated) - USB\VID_0BC2&PID_5031\NA0ENSYX]
>>>  Section start 2011/08/06 11:16:06.001
```

7)
Name NA0ENSYX
Last Written 08/08/11 08:36:20
Full Path Test 7 TS007 USB Analysis\TS007\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX

```
NA0ENSYX
```

Test 7 - EnCase Enscript Analysis

TS007 – Seagate GoFlex 500 GB USB 2.0/3.0 PSD Device\EnCase Enscript Analysis

---

The following information is from Test 7 TS007 USB Analysis_20110814 1559\TS007\Windows\System32\config\SYSTEM:
USBSTOR:

Type	Vendor	Product	Serial_Number	Friendly_Name	USB_Driver	Last_Written_Date	ParentIDPrefix
Disk	Seagate	FreeAgent_GoFlex	NA0ENSYX&0	Seagate FreeAgent GoFlex USB Device		08/08/11 08:27:05	NONE

\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}:

Type	Vendor	Product	Revision	Serial_Number	Driver	Last_Written_Date
Disk	Seagate	FreeAgent_GoFlex	_210	NA0ENSYX&0	{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	08/08/11 08:27:05

Mounted_Devices:

\DosDevices\F: DiskSignature: c5540530 VolumeByteOffsetStart: 32256

Summary Report: Test 7 - Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210 - FTK Registry Viewer PS Export

USBSTOR

Key Name	Name	Type	Data
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0		Key Properties	Last Written Time : 7/08/2011 20:27:05 UTC
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Capabilities	REG_DWORD	0x00000010 (16)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	HardwareID	REG_MULTI_SZ	USBSTOR\DiskSeagate_FreeAgent_GoFlex_210 USBSTOR\DiskSeagate_FreeAgent_GoFlex USBSTOR\DiskSeagate_ USBSTOR\Seagate_FreeAgent_GoFlex_ Seagate_FreeAgent_GoFlex_ USBSTOR\GenDisk GenDisk
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk USBSTOR\RAW

ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	ContainerID	REG_SZ	{2b43eb40-fe0e-5248-97bb-41455ca30041}
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\0013
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Class	REG_SZ	DiskDrive
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard disk drives)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Service	REG_SZ	disk
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	FriendlyName	REG_SZ	Seagate FreeAgent GoFlex USB Device

USB

Key Name	Name	Type	Data
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX		Key Properties	Last Written Time : 7/08/2011 20:36:20 UTC
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	DeviceDesc	REG_SZ	@usbstor.inf,%genericbulkonly.devicedesc%;USB Mass Storage Device
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	LocationInformation	REG_SZ	Port_#0003.Hub_#0005
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Capabilities	REG_DWORD	0x000000D4 (212)
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	HardwareID	REG_MULTI_SZ	USB\VID_0BC2&PID_5031&REV_0100 USB\VID_0BC2&PID_5031
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	CompatibleIDs	REG_MULTI_SZ	USB\Class_08&SubClass_06&Prot_50 USB\Class_08&SubClass_06

			USB\Class_08
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	ContainerID	REG_SZ	{2b43eb40-fe0e-5248-97bb-41455ca30041}
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	ClassGUID	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Driver	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}\0022
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Class	REG_SZ	USB
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Mfg	REG_SZ	@usbstor.inf,%generic.mfg%;Compatible USB storage device
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Service	REG_SZ	USBSTOR

DeviceClasses

Key Name	Name	Type	Data
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210#NA0ENSYX&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}		Key Properties	Last Written Time : 7/08/2011 20:27:05 UTC
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210#NA0ENSYX&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	DeviceInstance	REG_SZ	USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0

MountedDevices

Key Name	Name	Type	Data
MountedDevices		Key Properties	Last Written Time : 5/08/2011 23:16:11 UTC
MountedDevices	\DosDevices\F:	REG_BINARY	30 05 54 C5 00 7E 00 00 00 00 00 00
		(ASCII String)	0.T..~.....
		(UTF-16 String)	암緞

Summary Report: Test 7 - Seagate GoFlex - NTUSER.DAT - FTK Registry Viewer PSD Export

MountPoints2

Key Name	Name	Type	Data	
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{a2c76902-bfb8-11e0-825b-001fd0060cfd}\shell		Key Properties	Last Written Time	7/08/2011 20:36:21 UTC
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{a2c76902-bfb8-11e0-825b-001fd0060cfd}\shell	(default)	REG_SZ	None	

Summary Report: Test 7 – Seagate GoFlex – Setupapi.dev.log FTK 3.3 Bookmark Extracts

Name	setupapi.dev.log
Created Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Modified Date	8/08/2011 9:00:32 a.m. (2011-08-07 21:00:32 UTC)
Accessed Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Path	TS007.E01/NONAME [NTFS]/[root]/Windows/inf/setupapi.dev.log
Exported as	files\setupapi.dev.log.txt

>>> [Device Install (Hardware initiated) - USB\VID_0BC2&PID_5031\NA0ENSYX]
>>> Section start 2011/08/06 11:16:06.001

Test 6 – Seagate GoFlex USBlyzer Device Descriptor Details

USB Mass Storage Device

Connection Status Device connected
Current Configuration 1
Speed High
Device Address 1
Number Of Open Pipes 2

Device Descriptor FreeAgent GoFlex

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0210h	Invalid revision
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	40h	64 bytes
8	idVendor	2	0BC2h	Seagate RSS LLC
10	idProduct	2	5031h	
12	bcdDevice	2	0100h	1.00
14	iManufacturer	1	02h	"Seagate"
15	iProduct	1	03h	"FreeAgent GoFlex"
16	iSerialNumber	1	01h	"NA0ENSYX"
17	bNumConfigurations	1	01h	

Configuration Descriptor 1 Bus Powered, 500 mA

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	0020h	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	80h	Bus Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.0.....	No, Bus Powered
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	FAh	500 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

Endpoint Descriptor 02 2 Out, Bulk, 512 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0200h	512 bytes
6	bInterval	1	00h	

TEST 8 - Seagate 500 GB FreeAgent GoFlex USB 3.0 Portable Storage Device	
Test Details	Evidence Item: TS008 - Seagate 500 GB FreeAgent GoFlex USB 3.0 Portable Storage Device, Serial Number: NA0ENSYX. Connection to a Windows 7 Operating System for Evaluation and Evidence Analysis by the Sample Tools. USB 3.0 Supported PCIe Card Only. Identify any Additional Changes to the Windows 7 Registry When Using USB 3.0 Devices.
Tester	Mark Simms
Test Date(s)	15 August 2011
Conditional Requirements	**CR1** - The tool supports processing of digital source evidence (i.e. evidence file or individual "Live" and "Offline" Registry Hive data).
	CR2 - The tool supports date and time stamp reporting and or adjustment using the Coordinated Universal Time (UTC) time standard.
	CR3 - The tool supports the examination and reporting of USB 3.0 devices
	CR4 - All common Windows 7 Registry artifact and evidence groups are captured, processed and presented to a user by the tool.
	CR5 - All original digital source evidence is unchanged by any subsequent tool activity or user actions.
	CR6 - If processing errors occur whilst reading from the selected digital source, the tool displays an error notification to the user.
	CR7 - If the tool logs processing information, the information is accurately recorded in a log file or screen output to the user.
	CR8 - The tool allows extraction of analysis and log information into a format that is viewable and usable by the user.
Source and Destination Hard Drive Information	**Source:** Seagate ST3500418AS 500 GB Hard Drive, Serial Number: 5VMCLFM3. Sectors: 976,773,168. Interface: SATA. **Destination:** Western Digital 1.0 TB WD10EALX-009BA0 Hard Drive, Serial Number: WCATR4337006. Total Sectors: 1,953,525,169. Interface: SATA
Forensic Image & Hash Values	MD5 Hash Value: 8e210167914f5ad64b7d42cc00c01467. Created after the Tool 1 live state action. Relevant registry and system files hashed and exported.
Post Analysis Hash Value	MD5 Hash Value: 8e210167914f5ad64b7d42cc00c01467
Sample Toolset Details	Tool Name, Version, Developer Details, Usage or Licence Restrictions and Additional Software Requirements
Tool 1	USBDeview, v1.91, www.NirSoft.com, Freeware
Tool 2	USBDeviceForensics, v1.0.6, www.woanware.co.uk, Freeware
Tool 3	EnCase Forensic, v6.18, www.guidancesoftware.com, Commercial Licence, Additional EnScript: USB Device History, v 0.5 - Lance Muller, Freeware
Tool 4	FTK RegistryViewer, v1.6.3, www. accessdata.com, Commercial Licence. Used as a standalone application option within a licenced copy of Access Data's Forensic Tool Kit (FTK) v3.3 during the testing phase. A limited demonstration application is downloadable from the website but has no reporting or printing functionality.
Logging and Exported Data	Data output successfully exported for all of toolset examples except for Tool 1. Refer to attachments. Registry and system files hashed and exported.
Tool Results	All Conditional Requirements were met by Tools 2, 3 and 4.
Test Outcomes and Comments	The USB 3.0 PSD device was not detected by Tool 1. The Nirsoft website reports that the tool is not fully compatible with new USB 3.0 devices. No further testing could be completed using this tool. No additional Registry changes were noted when USB 3.0 devices are used on a Windows 7 operating system.

Test 8 USBDeviceForensics Seagate FreeAgent GoFlex PSD Export

---
Vendor: Ven_Seagate
Product: Prod_FreeAgent_GoFlex
Version: Rev__210
Serial No: NA0ENSYX
VID: VID_0BC2
PID: PID_5031
ParentIdPrefix:
Drive Letter:
Volume Name:
GUID:
MountPoint:
Install Date/Time): Monday, 1 January 0001 00:00:00
EMDMgmt Last Write Date/Time): Monday, 1 January 0001 00:00:00 UTC
First Time Connected After Last Reboot (USBSTOR Date/Time): Monday, 15 August 2011 04:08:06
First Time Connected After Last Reboot (DeviceClasses Date/Time): Monday, 1 January 0001 00:0
Last Time Connected (Enum\USB VIDPID Date/Time): Monday, 15 August 2011 04:08:06 UTC
Last Time Connected (MountPoints2 Date/Time):

Test 8 - EnCase Manual Analysis

TS008 - Seagate GoFlex 500 GB USB 3.0 PSD Device\EnCase Manual Analysis

1)
Name Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01
Last Written 05/08/2011 21:13:04
Full Path TS008 USB 3.0 GoFlex\TS008\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01

```
Disk&Ven_SanDisk&Prod_Cruzer&Rev_1.01
```

2)
Name NA0ENSYX&0
Last Written 15/08/2011 16:21:07
Full Path TS008 USB 3.0 GoFlex\TS008\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0

```
NA0ENSYX&0
```

3)
Name \DosDevices\F:
Last Written
Full Path TS008 USB 3.0 GoFlex\TS008\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\DosDevices\F:

```
300554C5007E0
```

4)
Name \??\Volume{078ccf70-c6f3-11e0-bc58-001fd0060cfd}
Last Written
Full Path TS008 USB 3.0 GoFlex\TS008\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\MountedDevices\\??\Volume{078ccf70-c6f3-11e0-bc58-001fd0060cfd}

```
300554C5007E0
```

5)
Name NA0ENSYX
Last Written 15/08/2011 16:21:07
Full Path TS008 USB 3.0 GoFlex\TS008\Windows\System32\config\SYSTEM\NTRegistry\CMI-CreateHive{F10156BE-0E87-4EFB-969E-5DA29D131144}\ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX

NA0ENSYX

6)
Name {078ccf70-c6f3-11e0-bc58-001fd0060cfd}
Last Written 15/08/2011 16:21:08
Full Path TS008 USB 3.0 GoFlex\TS008\Users\Test\NTUSER.DAT\NTRegistry\CMI-CreateHive{6A1C4018-979D-4291-A7DC-7AED1C75B67C}\Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{078ccf70-c6f3-11e0-bc58-001fd0060cfd}

```
{078ccf70-c6f3-11e0-bc58-001fd0060cfd}
```

Test 8 - EnCase Manual Analysis

7)
Name setupapi.dev.log
Last Written 15/08/2011 16:08:59
Full Path TS008 USB 3.0 GoFlex\TS008\Windows\inf\setupapi.dev.log

```
[Device Install (Hardware initiated) - USB\VID_0BC2&PID_5031\NA0ENSYX]
>>>  Section start 2011/
08/15 16:08:05.489
```

Test 8 - EnCase EnScript Analysis

TS008 – Seagate GoFlex 500 GB USB 3.0 PSD Device\EnCase Enscript Analysis

```
--------------------------------------------------------------------------------

The following information is from TS008 USB3.0 GoFlex\TS008\Windows\System32\config\SYSTEM:
USBSTOR:

Type   Vendor      Product     Serial_Number     Friendly_Name     USB_Driver        Last_Written_Date

Disk  Seagate     FreeAgent_GoFlex  NA0ENSYX&0  Seagate FreeAgent GoFlex USB Device  15/08/2011 16:21:07


ParentIDPrefix

NONE


\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}:

Type  Vendor      Product     Revision    Serial_Number                 Driver
      Last_Written_Date

Disk  Seagate     FreeAgent_GoFlex  _210  NA0ENSYX&0  {53f56307-b6bf-11d0-94f2-00a0c91efb8b}

15/08/2011 16:21:07


Mounted_Devices:

\DosDevices\F:    DiskSignature: c5540530 VolumeByteOffsetStart: 32256
```

Summary Report: Test 8 - Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210 - FTK Registry Viewer PSD Export

USBSTOR

Key Name	Name	Type	Data
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0		Key Properties	Last Written Time : 15/08/2011 4:08:06 UTC
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	DeviceDesc	REG_SZ	@disk.inf,%disk_devdesc%;Disk drive
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Capabilities	REG_DWORD	0x00000010 (16)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	UINumber	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	HardwareID	REG_MULTI_SZ	USBSTOR\DiskSeagate_FreeAgent_GoFlex_210 USBSTOR\DiskSeagate_FreeAgent_GoFlex USBSTOR\DiskSeagate_ USBSTOR\Seagate_FreeAgent_GoFlex_ Seagate_FreeAgent_GoFlex_ USBSTOR\GenDisk GenDisk
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	CompatibleIDs	REG_MULTI_SZ	USBSTOR\Disk USBSTOR\RAW
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	ContainerID	REG_SZ	{2b43eb40-fe0e-5248-97bb-41455ca30041}
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	ClassGUID	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Driver	REG_SZ	{4d36e967-e325-11ce-bfc1-08002be10318}\0009
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Class	REG_SZ	DiskDrive
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Mfg	REG_SZ	@disk.inf,%genmanufacturer%;(Standard disk drives)
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	Service	REG_SZ	disk
ControlSet001\Enum\USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0	FriendlyName	REG_SZ	Seagate FreeAgent GoFlex USB Device

USB

Key Name	Name	Type	Data
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX		Key Properties	Last Written Time : 15/08/2011 4:08:06 UTC
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	DeviceDesc	REG_SZ	@usbstor.inf,%genericbulkonly.devicedesc%;USB Mass Storage Device
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	LocationInformation	REG_SZ	Port_#0002.Hub_#0001
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Capabilities	REG_DWORD	0x000000D4 (212)
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	UINumber	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	HardwareID	REG_MULTI_SZ	USB\VID_0BC2&PID_5031&REV_0100
			USB\VID_0BC2&PID_5031
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	CompatibleIDs	REG_MULTI_SZ	USB\Class_08&SubClass_06&Prot_50
			USB\Class_08&SubClass_06
			USB\Class_08
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	ContainerID	REG_SZ	{2b43eb40-fe0e-5248-97bb-41455ca30041}
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	ConfigFlags	REG_DWORD	0x00000000 (0)
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	ClassGUID	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Driver	REG_SZ	{36fc9e60-c465-11cf-8056-444553540000}\0020
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Class	REG_SZ	USB
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Mfg	REG_SZ	@usbstor.inf,%generic.mfg%;Compatible USB storage device
ControlSet001\Enum\USB\VID_0BC2&PID_5031\NA0ENSYX	Service	REG_SZ	USBSTOR

DeviceClasses

Key Name	Name	Type	Data
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210#NA0ENSYX&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}		Key Properties	Last Written Time : 15/08/2011 4:08:06 UTC
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210#NA0ENSYX&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	DeviceInstance	REG_SZ	USBSTOR\Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210\NA0ENSYX&0
ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210#NA0ENSYX&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\#		Key Properties	Last Written Time : 15/08/2011 4:08:06 UTC

ControlSet001\Control\DeviceClasses\{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\##?#USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210#NA0ENSYX&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}\#		SymbolicLink	REG_SZ	\\?\USBSTOR#Disk&Ven_Seagate&Prod_FreeAgent_GoFlex&Rev__210#NA0ENSYX&0#{53f56307-b6bf-11d0-94f2-00a0c91efb8b}	

MountedDevices

Key Name	Name	Type	Data
MountedDevices		Key Properties	Last Written Time : 15/08/2011 4:08:07 UTC
MountedDevices	\DosDevices\F:	REG_BINARY	30 05 54 C5 00 7E 00 00 00 00 00 00
		(ASCII String)	0.T..~.....
		(UTF-16 String)	0암緞

Summary Report: Test 8 - Seagate GoFlex - NTUSER.DAT - FTK Registry Viewer PSD Export

MountPoints2

Key Name	Name	Type	Data	
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{078ccf70-c6f3-11e0-bc58-001fd0060cfd}\shell		Key Properties	Last Written Time	: 15/08/2011 4:08:08 UTC
Software\Microsoft\Windows\CurrentVersion\Explorer\MountPoints2\{078ccf70-c6f3-11e0-bc58-001fd0060cfd}\shell	(default)	REG_SZ	None	

Summary Report: Test 8 – Seagate GoFlex – Setupapi.dev.log FTK 3.3 Bookmark Extracts

Name	setupapi.dev.log
Created Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Modified Date	15/08/2011 4:08:59 p.m. (2011-08-15 04:08:59 UTC)
Accessed Date	5/08/2011 5:32:35 p.m. (2011-08-05 05:32:35 UTC)
Path	TS008.E01/NONAME [NTFS]/[root]/Windows/inf/setupapi.dev.log
Exported as	files\setupapi.dev.log.txt

>>> [Device Install (Hardware initiated) - USB\VID_0BC2&PID_5031\NA0ENSYX]
>>> Section start 2011/08/15 16:08:05.489

Test 8 – Seagate GoFlex USBlyzer Device Descriptor Details

USB Mass Storage Device

Connection Status Device connected
Current Configuration 1
Speed Unknown
Device Address 1
Number Of Open Pipes 2
Device Descriptor FreeAgent GoFlex

Offset	Field	Size	Value	Description
0	bLength	1	12h	
1	bDescriptorType	1	01h	Device
2	bcdUSB	2	0300h	Invalid revision
4	bDeviceClass	1	00h	Class info in Ifc Descriptors
5	bDeviceSubClass	1	00h	
6	bDeviceProtocol	1	00h	
7	bMaxPacketSize0	1	09h	Should be 8, 16, 32, or 64 bytes
8	idVendor	2	0BC2h	Seagate RSS LLC
10	idProduct	2	5031h	
12	bcdDevice	2	0100h	1.00
14	iManufacturer	1	02h	"Seagate"
15	iProduct	1	03h	"FreeAgent GoFlex"
16	iSerialNumber	1	01h	"NA0ENSYX"
17	bNumConfigurations	1	01h	

Configuration Descriptor 1 Self Powered

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	02h	Configuration
2	wTotalLength	2	002Ch	
4	bNumInterfaces	1	01h	
5	bConfigurationValue	1	01h	
6	iConfiguration	1	00h	
7	bmAttributes	1	C0h	Self Powered
	4..0: Reserved		...00000	
	5: Remote Wakeup		..0.....	No
	6: Self Powered		.1......	Yes
	7: Reserved (set to one) (bus-powered for 1.0)		1.......	
8	bMaxPower	1	12h	36 mA

Interface Descriptor 0/0 Mass Storage, 2 Endpoints

Offset	Field	Size	Value	Description
0	bLength	1	09h	
1	bDescriptorType	1	04h	Interface
2	bInterfaceNumber	1	00h	
3	bAlternateSetting	1	00h	
4	bNumEndpoints	1	02h	
5	bInterfaceClass	1	08h	Mass Storage
6	bInterfaceSubClass	1	06h	SCSI Transparent Command Set
7	bInterfaceProtocol	1	50h	Bulk-Only Transport
8	iInterface	1	00h	

Endpoint Descriptor 81 1 In, Bulk, 1024 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	81h	1 In
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0400h	1024 bytes
6	bInterval	1	00h	

Unrecognized Class-Specific Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	06h	
1	bDescriptorType	1	30h	
2		4	0F 00 00 00	

Endpoint Descriptor 02 2 Out, Bulk, 1024 bytes

Offset	Field	Size	Value	Description
0	bLength	1	07h	
1	bDescriptorType	1	05h	Endpoint
2	bEndpointAddress	1	02h	2 Out
3	bmAttributes	1	02h	Bulk
	1..0: Transfer Type		10	Bulk
	7..2: Reserved		000000..	
4	wMaxPacketSize	2	0400h	1024 bytes
6	bInterval	1	00h	

Unrecognized Class-Specific Descriptor

Offset	Field	Size	Value	Description
0	bLength	1	06h	
1	bDescriptorType	1	30h	
2		4	0F 00 00 00	

Appendix D – Prototype Tool Testing Reports

Case Data

Case or Matter Number:	DT001
Exhibit or Item Number:	Not Assigned
Analysis Date/Time:	Sunday, 04/12/2011 16:09:30 (UTC Sunday, 04/12/2011 03:09:30)
Investigator Name or ID:	Not Assigned
Analyst Name or ID:	Mark Simms
Exhibit Notes:	SanDisk 4 GB Cruzer USB 2.0 Flash Drive Serial Number: 2005304502028AB1BCA4 Suspect Evidence Set: TS001 Connection: USB 2.0 Only
Computer Name:	TEST-PC
Windows Product Name:	Windows 7 Home Premium Version 6.1
Account Username:	Test

(HTML Report Printed to Microsoft Word for Appendix Use Only)

USB Device Details

Device Number:	1
VendorID:	058F
Product ID:	6377
Version:	1.01
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB CF Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&1
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487c-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.849
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 17:33:53 (UTC Friday, 05/08/2011 05:33:53)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	2
VendorID:	058F
Product ID:	6377
Version:	1.03
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB MS Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&3
Mounted Devices: Last Drive Letter Mapping:	H:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487e-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:20.644
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	H:\ Friday, 05/08/2011 17:34:00 (UTC Friday, 05/08/2011 05:34:00)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	3
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SD Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&0
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487b-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:18.944
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	E:\ Friday, 05/08/2011 17:33:50 (UTC Friday, 05/08/2011 05:33:50)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	4
VendorID:	058F
Product ID:	6377
Version:	1.02
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SM Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&2
Mounted Devices: Last Drive Letter Mapping:	G:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487d-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.303
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	G:\ Friday, 05/08/2011 17:33:55 (UTC Friday, 05/08/2011 05:33:55)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	5
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	HP Photosmart C4180 USB Device
Class:	DiskDrive
Device Serial Number:	7&7949063&0&MY6C2H506V04J7&0
Mounted Devices: Last Drive Letter Mapping:	I:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d034881-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:34:02.468
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	I:\ Friday, 05/08/2011 17:34:05 (UTC Friday, 05/08/2011 05:34:05)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Device Alert Status:	No matches

Device Number:	6
VendorID:	0781
Product ID:	5530
Version:	1.01
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	SanDisk Cruzer USB Device
Class:	DiskDrive
Device Serial Number:	2005304502028AB1BCA4&0
Mounted Devices: Last Drive Letter Mapping or Disk Signature:	F:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{ebda7496-bf42-11e0-b600-001fd0060cfd}
User Account NTUSER.DAT: MountPoints2:	Object was used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 21:13:00.716
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 21:13:05 (UTC Friday, 05/08/2011 09:13:05)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 21:13:08 (UTC Friday, 05/08/2011 09:13:08)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 21:13:05 (UTC Friday, 05/08/2011 09:13:05)
Device Alert Status:	ALERT: DEVICE FOUND

Completion date/time Sunday, 04/12/2011 16:09:32
(UTC Sunday, 04/12/2011 03:09:32)
Total Devices Found: 6
Device Alerts: 1 (Last one on device 6)
Processing time 1.62 secs

Report Notes

Device Alerts
=========

The Device Alert Status field count in this report represents the number of times identical USB device descriptor artifacts were found in the suspect evidence set and then co-located in the registry files of the analyst workstation.

This program employs a comparison analysis methodology to make a determination as to the common origin of items of evidence, by utilising registry file entries from the suspect evidence set and registry file entries of the analyst workstation after the suspect USB device had been connected to it.

Write-blocking precautions are utilised as a matter of course when the suspect USB device is connected to the analyst workstation during examination.

When USB devices are connected to a computer system, they leave unique signatures and associated entries in various system and registry files of the Windows Operating System.

Each unique "signature" can be searched for in the recovered suspect workstation registry files.

"No matches" appearing in the Alert field would indicate that the suspect USB device has not previously been connected to the suspect computer system whilst an "Alert" would indicate that the suspect USB device has previously been connected to the suspect computer by a user.

Results
=====

A blank "Mounted Devices" key field, would indicate that the drive letter was reused by another device or there was no information recorded for extraction.

An external USB hard drive and or traditional hard drive will populate the “Mounted Devices” key field with a hexadecimal disk signature value.

The "Windows Portable Devices" key field is not populated by external USB hard drive or drive letter information. Other USB devices may be recorded in this sub-key location of the SOFTWARE hive file.

This report is produced pursuant to S137(1) & S137(2) New Zealand Evidence Act 2006: "Evidence produced by machine, device, or technical process"

END REPORT

USB Examination Report

Case Data

Case or Matter Number:	DT002
Exhibit or Item Number:	Not Assigned
Analysis Date/Time:	Sunday, 04/12/2011 16:15:40 (UTC Sunday, 04/12/2011 03:15:40)
Investigator Name or ID:	Not Assigned
Analyst Name or ID:	Mark Simms
Exhibit Notes:	Kingston DT101 4 GB Data Traveler 101 USB 2.0 Flash Drive Serial Number: 001A4D5F1A5CBB11200012D6 Suspect Evidence Set: TS002 Connection: USB 2.0 Only
Computer Name:	TEST-PC
Windows Product Name:	Windows 7 Home Premium Version 6.1
Account Username:	Test

(HTML Report Printed to Microsoft Word for Appendix Use Only)

USB Device Details

Device Number:	1
VendorID:	058F
Product ID:	6377
Version:	1.01
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB CF Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&1
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487c-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.849
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 17:33:53 (UTC Friday, 05/08/2011 05:33:53)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	2
VendorID:	058F
Product ID:	6377
Version:	1.03
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB MS Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&3
Mounted Devices: Last Drive Letter Mapping:	H:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487e-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:20.644
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	H:\ Friday, 05/08/2011 17:34:00 (UTC Friday, 05/08/2011 05:34:00)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	3
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SD Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&0
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487b-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:18.944
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	E:\ Friday, 05/08/2011 17:33:50 (UTC Friday, 05/08/2011 05:33:50)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	4
VendorID:	058F
Product ID:	6377
Version:	1.02
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SM Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&2
Mounted Devices: Last Drive Letter Mapping:	G:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487d-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.303
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	G:\ Friday, 05/08/2011 17:33:55 (UTC Friday, 05/08/2011 05:33:55)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	5
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	HP Photosmart C4180 USB Device
Class:	DiskDrive
Device Serial Number:	7&7949063&0&MY6C2H506V04J7&0
Mounted Devices: Last Drive Letter Mapping:	I:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d034881-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:34:02.468
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	I:\ Friday, 05/08/2011 17:34:05 (UTC Friday, 05/08/2011 05:34:05)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Device Alert Status:	No matches

Device Number:	6
VendorID:	0951
Product ID:	1642
Version:	PMAP
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Kingston DT 101 G2 USB Device
Class:	DiskDrive
Device Serial Number:	001A4D5F1A5CBB11200012D6&0
Mounted Devices: Last Drive Letter Mapping or Disk Signature:	F:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{ebda74c0-bf42-11e0-b600-001fd0060cfd}
User Account NTUSER.DAT: MountPoints2:	Object was used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 22:00:00.571
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 22:00:03 (UTC Friday, 05/08/2011 10:00:03)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	KINGSTON Friday, 05/08/2011 22:00:05 (UTC Friday, 05/08/2011 10:00:05)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 22:00:03 (UTC Friday, 05/08/2011 10:00:03)
Device Alert Status:	ALERT: DEVICE FOUND

Completion date/time Sunday, 04/12/2011 16:15:41 (UTC Sunday, 04/12/2011 03:15:41)
Total Devices Found: 6
Device Alerts: 1 (Last one on device 6)
Processing time 1.10 secs

Report Notes

Device Alerts
=========

The Device Alert Status field count in this report represents the number of times identical USB device descriptor artifacts were found in the suspect evidence set and then co-located in the registry files of the analyst workstation.

This program employs a comparison analysis methodology to make a determination as to the common origin of items of evidence, by utilising registry file entries from the suspect evidence set and registry file entries of the analyst workstation after the suspect USB device had been connected to it.

Write-blocking precautions are utilised as a matter of course when the suspect USB device is connected to the analyst workstation during examination.

When USB devices are connected to a computer system, they leave unique signatures and associated entries in various system and registry files of the Windows Operating System.

Each unique "signature" can be searched for in the recovered suspect workstation registry files.

"No matches" appearing in the Alert field would indicate that the suspect USB device has not previously been connected to the suspect computer system whilst an "Alert" would indicate that the suspect USB device has previously been connected to the suspect computer by a user.

Results
=====

A blank "Mounted Devices" key field, would indicate that the drive letter was reused by another device or there was no information recorded for extraction.

An external USB hard drive and or traditional hard drive will populate the “Mounted Devices” key field with a hexadecimal disk signature value.

The "Windows Portable Devices" key field is not populated by external USB hard drive or drive letter information. Other USB devices may be recorded in this sub-key location of the SOFTWARE hive file.

This report is produced pursuant to S137(1) & S137(2) New Zealand Evidence Act 2006: "Evidence produced by machine, device, or technical process"

END REPORT

USB Examination Report

Case Data

Case or Matter Number:	DT003
Exhibit or Item Number:	Not Assigned
Analysis Date/Time:	Sunday, 04/12/2011 16:19:57 (UTC Sunday, 04/12/2011 03:19:57)
Investigator Name or ID:	Not Assigned
Analyst Name or ID:	Mark Simms
Exhibit Notes:	Apacer AH3255 4 GB USB 2.0 Flash Drive Serial Number: 000FF1103192249410006123 Suspect Evidence Set: TS003 Connection: USB 2.0 Only
Suspect Computer Name:	TEST-PC
Windows Product Name:	Windows 7 Home Premium Version 6.1
Account Username:	Test

(HTML Report Printed to Microsoft Word for Appendix Use Only)

USB Device Details

Device Number:	1
VendorID:	058F
Product ID:	6377
Version:	1.01
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB CF Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&1
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487c-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.849
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 17:33:53 (UTC Friday, 05/08/2011 05:33:53)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	2
VendorID:	058F
Product ID:	6377
Version:	1.03
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB MS Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&3
Mounted Devices: Last Drive Letter Mapping:	H:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487e-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:20.644
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	H:\ Friday, 05/08/2011 17:34:00 (UTC Friday, 05/08/2011 05:34:00)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	3
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SD Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&0
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487b-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:18.944
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	E:\ Friday, 05/08/2011 17:33:50 (UTC Friday, 05/08/2011 05:33:50)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	4
VendorID:	058F
Product ID:	6377
Version:	1.02
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SM Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&2
Mounted Devices: Last Drive Letter Mapping:	G:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487d-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.303
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	G:\ Friday, 05/08/2011 17:33:55 (UTC Friday, 05/08/2011 05:33:55)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	5
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	HP Photosmart C4180 USB Device
Class:	DiskDrive
Device Serial Number:	7&7949063&0&MY6C2H506V04J7&0
Mounted Devices: Last Drive Letter Mapping:	I:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d034881-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:34:02.468
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	I:\ Friday, 05/08/2011 17:34:05 (UTC Friday, 05/08/2011 05:34:05)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Device Alert Status:	No matches

Device Number:	6
VendorID:	1005
Product ID:	B113
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	USB FLASH DRIVE USB Device
Class:	DiskDrive
Device Serial Number:	000FF1103192249410006123&0
Mounted Devices: Last Drive Letter Mapping or Disk Signature:	F:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{ebda74df-bf42-11e0-b600-001fd0060cfd}
User Account NTUSER.DAT: MountPoints2:	Object was used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 22:41:29.684
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 22:41:32 (UTC Friday, 05/08/2011 10:41:32)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 22:41:36 (UTC Friday, 05/08/2011 10:41:36)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 22:41:32 (UTC Friday, 05/08/2011 10:41:32)
Device Alert Status:	ALERT: DEVICE FOUND

Completion date/time Sunday, 04/12/2011 16:19:58
(UTC Sunday, 04/12/2011 03:19:58)
Total Devices Found: 6
Device Alerts: 1 (Last one on device 6)
Processing time 1.24 secs

Report Notes

Device Alerts
=========

The Device Alert Status field count in this report represents the number of times identical USB device descriptor artifacts were found in the suspect evidence set and then co-located in the registry files of the analyst workstation.

This program employs a comparison analysis methodology to make a determination as to the common origin of items of evidence, by utilising registry file entries from the suspect evidence set and registry file entries of the analyst workstation after the suspect USB device had been connected to it.

Write-blocking precautions are utilised as a matter of course when the suspect USB device is connected to the analyst workstation during examination.

When USB devices are connected to a computer system, they leave unique signatures and associated entries in various system and registry files of the Windows Operating System.

Each unique "signature" can be searched for in the recovered suspect workstation registry files.

"No matches" appearing in the Alert field would indicate that the suspect USB device has not previously been connected to the suspect computer system whilst an "Alert" would indicate that the suspect USB device has previously been connected to the suspect computer by a user.

Results
=====

A blank "Mounted Devices" key field, would indicate that the drive letter was reused by another device or there was no information recorded for extraction.

An external USB hard drive and or traditional hard drive will populate the “Mounted Devices” key field with a hexadecimal disk signature value.

The "Windows Portable Devices" key field is not populated by external USB hard drive or drive letter information. Other USB devices may be recorded in this sub-key location of the SOFTWARE hive file.

This report is produced pursuant to S137(1) & S137(2) New Zealand Evidence Act 2006: "Evidence produced by machine, device, or technical process"

END REPORT

USB Examination Report

Case Data

Case or Matter Number:	DT004
Exhibit or Item Number:	Not Assigned
Analysis Date/Time:	Sunday, 04/12/2011 16:28:28 (UTC Sunday, 04/12/2011 03:28:28)
Investigator Name or ID:	Not Assigned
Analyst Name or ID:	Mark Simms
Exhibit Notes:	Dick Smith 2 GB USB 2.0 Micro Drive Serial Number: C7E69A7C Suspect Evidence Set: TS004 Connection: USB 2.0 Only
Suspect Computer Name:	TEST-PC
Windows Product Name:	Windows 7 Home Premium Version 6.1
Account Username:	Test

(HTML Report Printed to Microsoft Word for Appendix Use Only)

USB Device Details

Device Number:	1
VendorID:	058F
Product ID:	6387
Version:	8.07
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	DS MicroDrive 2GB USB Device
Class:	DiskDrive
Device Serial Number:	C7E69A7C&0
Mounted Devices: Last Drive Letter Mapping:	F:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{ebda74f5-bf42-11e0-b600-001fd0060cfd}
User Account NTUSER.DAT: MountPoints2:	Object was used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 23:22:59.179
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 23:23:02 (UTC Friday, 05/08/2011 11:23:02)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 23:23:05 (UTC Friday, 05/08/2011 11:23:05)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 23:23:02 (UTC Friday, 05/08/2011 11:23:02)
Device Alert Status:	ALERT: DEVICE FOUND

Device Number:	2
VendorID:	058F
Product ID:	6377
Version:	1.01
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB CF Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&1
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487c-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.849
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 17:33:53 (UTC Friday, 05/08/2011 05:33:53)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	3
VendorID:	058F
Product ID:	6377
Version:	1.03
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB MS Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&3
Mounted Devices: Last Drive Letter Mapping:	H:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487e-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:20.644
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	H:\ Friday, 05/08/2011 17:34:00 (UTC Friday, 05/08/2011 05:34:00)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	4
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SD Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&0
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487b-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:18.944
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	E:\ Friday, 05/08/2011 17:33:50 (UTC Friday, 05/08/2011 05:33:50)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	5
VendorID:	058F
Product ID:	6377
Version:	1.02
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SM Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&2
Mounted Devices: Last Drive Letter Mapping:	G:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487d-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.303
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	G:\ Friday, 05/08/2011 17:33:55 (UTC Friday, 05/08/2011 05:33:55)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	6
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	HP Photosmart C4180 USB Device
Class:	DiskDrive
Device Serial Number:	7&7949063&0&MY6C2H506V04J7&0
Mounted Devices: Last Drive Letter Mapping:	I:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d034881-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:34:02.468
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	I:\ Friday, 05/08/2011 17:34:05 (UTC Friday, 05/08/2011 05:34:05)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Device Alert Status:	No matches

Completion date/time Sunday, 04/12/2011 16:28:29
(UTC Sunday, 04/12/2011 03:28:29)
Total Devices Found: 6
Device Alerts: 1 (Last one on device 1)
Processing time 1.31 secs

Report Notes

Device Alerts
=========

The Device Alert Status field count in this report represents the number of times identical USB device descriptor artifacts were found in the suspect evidence set and then co-located in the registry files of the analyst workstation.

This program employs a comparison analysis methodology to make a determination as to the common origin of items of evidence, by utilising registry file entries from the suspect evidence set and registry file entries of the analyst workstation after the suspect USB device had been connected to it.

Write-blocking precautions are utilised as a matter of course when the suspect USB device is connected to the analyst workstation during examination.

When USB devices are connected to a computer system, they leave unique signatures and associated entries in various system and registry files of the Windows Operating System.

Each unique "signature" can be searched for in the recovered suspect workstation registry files.

"No matches" appearing in the Alert field would indicate that the suspect USB device has not previously been connected to the suspect computer system whilst an "Alert" would indicate that the suspect USB device has previously been connected to the suspect computer by a user.

Results
=====

A blank "Mounted Devices" key field, would indicate that the drive letter was reused by another device or there was no information recorded for extraction.

An external USB hard drive and or traditional hard drive will populate the “Mounted Devices” key field with a hexadecimal disk signature value.

The "Windows Portable Devices" key field is not populated by external USB hard drive or drive letter information. Other USB devices may be recorded in this sub-key location of the SOFTWARE hive file.

This report is produced pursuant to S137(1) & S137(2) New Zealand Evidence Act 2006: "Evidence produced by machine, device, or technical process"

END REPORT

USB Examination Report

Case Data

Case or Matter Number:	DT005
Exhibit or Item Number:	Not Assigned
Analysis Date/Time:	Sunday, 04/12/2011 16:36:32 (UTC Sunday, 04/12/2011 03:36:32)
Investigator Name or ID:	Not Assigned
Analyst Name or ID:	Mark Simms
Exhibit Notes:	Transcend StoreJet 500 USB 2.0 Portable Storage Device Serial Number: 1A9306339FFF Suspect Evidence Set: TS005 Connection: USB 2.0 Only
Suspect Computer Name:	TEST-PC
Windows Product Name:	Windows 7 Home Premium Version 6.1
Account Username:	Test

(HTML Report Printed to Microsoft Word for Appendix Use Only)

USB Device Details

Device Number:	1
VendorID:	058F
Product ID:	6377
Version:	1.01
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB CF Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&1
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487c-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.849
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 17:33:53 (UTC Friday, 05/08/2011 05:33:53)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	2
VendorID:	058F
Product ID:	6377
Version:	1.03
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB MS Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&3
Mounted Devices: Last Drive Letter Mapping:	H:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487e-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:20.644
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	H:\ Friday, 05/08/2011 17:34:00 (UTC Friday, 05/08/2011 05:34:00)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	3
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SD Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&0
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487b-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:18.944
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	E:\ Friday, 05/08/2011 17:33:50 (UTC Friday, 05/08/2011 05:33:50)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	4
VendorID:	058F
Product ID:	6377
Version:	1.02
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SM Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&2
Mounted Devices: Last Drive Letter Mapping:	G:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487d-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.303
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	G:\ Friday, 05/08/2011 17:33:55 (UTC Friday, 05/08/2011 05:33:55)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	5
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	HP Photosmart C4180 USB Device
Class:	DiskDrive
Device Serial Number:	7&7949063&0&MY6C2H506V04J7&0
Mounted Devices: Last Drive Letter Mapping:	I:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d034881-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:34:02.468
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	I:\ Friday, 05/08/2011 17:34:05 (UTC Friday, 05/08/2011 05:34:05)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Device Alert Status:	No matches

Device Number:	6
VendorID:	152D
Product ID:	2509
Version:	Rev_0100
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	StoreJet Transcend USB Device
Class:	DiskDrive
Device Serial Number:	1A9306339FFF&0
Mounted Devices: Last Drive Letter Mapping or Disk Signature:	F: 6B 35 05 0E
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{ebda750c-bf42-11e0-b600-001fd0060cfd}
User Account NTUSER.DAT: MountPoints2:	Object was used by this user
Setupapi.dev.log: First Device Connection Date/Time:	06/08/2011 00:02:04.347
DeviceClasses: First Connection Date/Time After Reboot:	Saturday, 06/08/2011 00:02:06 (UTC Friday, 05/08/2011 12:02:06)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	No entry for external USB drive
USBSTOR: Last Written Date/Time Stamp:	Saturday, 06/08/2011 00:02:06 (UTC Friday, 05/08/2011 12:02:06)
Device Alert Status:	ALERT: DEVICE FOUND

Completion date/time Sunday, 04/12/2011 16:36:34
(UTC Sunday, 04/12/2011 03:36:34)
Total Devices Found: 6
Device Alerts: 1 (Last one on device 6)
Processing time 1.45 secs

Report Notes

Device Alerts
=========

The Device Alert Status field count in this report represents the number of times identical USB device descriptor artifacts were found in the suspect evidence set and then co-located in the registry files of the analyst workstation.

This program employs a comparison analysis methodology to make a determination as to the common origin of items of evidence, by utilising registry file entries from the suspect evidence set and registry file entries of the analyst workstation after the suspect USB device had been connected to it.

Write-blocking precautions are utilised as a matter of course when the suspect USB device is connected to the analyst workstation during examination.

When USB devices are connected to a computer system, they leave unique signatures and associated entries in various system and registry files of the Windows Operating System.

Each unique "signature" can be searched for in the recovered suspect workstation registry files.

"No matches" appearing in the Alert field would indicate that the suspect USB device has not previously been connected to the suspect computer system whilst an "Alert" would indicate that the suspect USB device has previously been connected to the suspect computer by a user.

Results
=====

A blank "Mounted Devices" key field, would indicate that the drive letter was reused by another device or there was no information recorded for extraction.

An external USB hard drive and or traditional hard drive will populate the “Mounted Devices” key field with a hexadecimal disk signature value.

The "Windows Portable Devices" key field is not populated by external USB hard drive or drive letter information. Other USB devices may be recorded in this sub-key location of the SOFTWARE hive file.

This report is produced pursuant to S137(1) & S137(2) New Zealand Evidence Act 2006: "Evidence produced by machine, device, or technical process"

END REPORT

USB Examination Report

Case Data

Case or Matter Number:	DT006
Exhibit or Item Number:	Not Assigned
Analysis Date/Time:	Sunday, 04/12/2011 16:43:18 (UTC Sunday, 04/12/2011 03:43:18)
Investigator Name or ID:	Not Assigned
Analyst Name or ID:	Mark Simms
Exhibit Notes:	Seagate 500 GB FreeAgent GoFlex USB 2.0/3.0 Portable Storage Device Serial Number: NA0ENSYX Suspect Evidence Set: TS006 Connection: USB 2.0 Only
Suspect Computer Name:	TEST-PC
Windows Product Name:	Windows 7 Home Premium Version 6.1
Account Username:	Test

(HTML Report Printed to Microsoft Word for Appendix Use Only)

USB Device Details

Device Number:	1
VendorID:	058F
Product ID:	6377
Version:	1.01
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB CF Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&1
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487c-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.849
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 17:33:53 (UTC Friday, 05/08/2011 05:33:53)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	2
VendorID:	058F
Product ID:	6377
Version:	1.03
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB MS Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&3
Mounted Devices: Last Drive Letter Mapping:	H:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487e-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:20.644
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	H:\ Friday, 05/08/2011 17:34:00 (UTC Friday, 05/08/2011 05:34:00)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	3
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SD Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&0
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487b-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:18.944
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	E:\ Friday, 05/08/2011 17:33:50 (UTC Friday, 05/08/2011 05:33:50)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	4
VendorID:	058F
Product ID:	6377
Version:	1.02
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SM Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&2
Mounted Devices: Last Drive Letter Mapping:	G:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487d-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.303
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	G:\ Friday, 05/08/2011 17:33:55 (UTC Friday, 05/08/2011 05:33:55)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	5
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	HP Photosmart C4180 USB Device
Class:	DiskDrive
Device Serial Number:	7&7949063&0&MY6C2H506V04J7&0
Mounted Devices: Last Drive Letter Mapping:	I:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d034881-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:34:02.468
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	I:\ Friday, 05/08/2011 17:34:05 (UTC Friday, 05/08/2011 05:34:05)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Device Alert Status:	No matches

Device Number:	6
VendorID:	0BC2
Product ID:	5031
Version:	Rev_210
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Seagate FreeAgent GoFlex USB Device
Class:	DiskDrive
Device Serial Number:	NA0ENSYX&0
Mounted Devices: Last Drive Letter Mapping or Disk Signature:	F: 30 05 54 C5
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{a2c76902-bfb8-11e0-825b-001fd0060cfd}
User Account NTUSER.DAT: MountPoints2:	Object was used by this user
Setupapi.dev.log: First Device Connection Date/Time:	06/08/2011 11:16:06.001
DeviceClasses: First Connection Date/Time After Reboot:	Saturday, 06/08/2011 11:16:10 (UTC Friday, 05/08/2011 23:16:10)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	No entry for external USB drive
USBSTOR: Last Written Date/Time Stamp:	Saturday, 06/08/2011 11:16:10 (UTC Friday, 05/08/2011 23:16:10)
Device Alert Status:	ALERT: DEVICE FOUND

Completion date/time Sunday, 04/12/2011 16:43:19
(UTC Sunday, 04/12/2011 03:43:19)
Total Devices Found: 6
Device Alerts: 1 (Last one on device 6)
Processing time 1.75 secs

Report Notes

Device Alerts
=========

The Device Alert Status field count in this report represents the number of times identical USB device descriptor artifacts were found in the suspect evidence set and then co-located in the registry files of the analyst workstation.

This program employs a comparison analysis methodology to make a determination as to the common origin of items of evidence, by utilising registry file entries from the suspect evidence set and registry file entries of the analyst workstation after the suspect USB device had been connected to it.

Write-blocking precautions are utilised as a matter of course when the suspect USB device is connected to the analyst workstation during examination.

When USB devices are connected to a computer system, they leave unique signatures and associated entries in various system and registry files of the Windows Operating System.

Each unique "signature" can be searched for in the recovered suspect workstation registry files.

"No matches" appearing in the Alert field would indicate that the suspect USB device has not previously been connected to the suspect computer system whilst an "Alert" would indicate that the suspect USB device has previously been connected to the suspect computer by a user.

Results
=====

A blank "Mounted Devices" key field, would indicate that the drive letter was reused by another device or there was no information recorded for extraction.

An external USB hard drive and or traditional hard drive will populate the “Mounted Devices” key field with a hexadecimal disk signature value.

The "Windows Portable Devices" key field is not populated by external USB hard drive or drive letter information. Other USB devices may be recorded in this sub-key location of the SOFTWARE hive file.

This report is produced pursuant to S137(1) & S137(2) New Zealand Evidence Act 2006: "Evidence produced by machine, device, or technical process"

END REPORT

USB Examination Report

Case Data

Case or Matter Number:	DT008
Exhibit or Item Number:	Not Assigned
Analysis Date/Time:	Friday, 09/12/2011 12:52:45 (UTC Thursday, 08/12/2011 23:52:45)
Investigator Name or ID:	Not Assigned
Analyst Name or ID:	Mark Simms
Exhibit Notes:	Seagate FreeAgent GoFlex USB 2.0/3.0 USB Portable Storage Device Serial Number: NA0ENSYX Capacity: 500 GB Suspect Evidence Set: TS008 Connection: USB 3.0 Only
Suspect Computer Name:	TEST-PC
Windows Product Name:	Windows 7 Home Premium Version 6.1
Account Username:	Test

(HTML Report Printed to Microsoft Word for Appendix Use Only)

USB Device Details

Device Number:	1
VendorID:	058F
Product ID:	6377
Version:	1.01
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB CF Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&1
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487c-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.849
DeviceClasses: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	F:\ Friday, 05/08/2011 17:33:53 (UTC Friday, 05/08/2011 05:33:53)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	2
VendorID:	058F
Product ID:	6377
Version:	1.03
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB MS Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&3
Mounted Devices: Last Drive Letter Mapping:	H:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487e-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:20.644
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	H:\ Friday, 05/08/2011 17:34:00 (UTC Friday, 05/08/2011 05:34:00)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	3
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SD Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&0
Mounted Devices: Last Drive Letter Mapping:	
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487b-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:18.944
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	E:\ Friday, 05/08/2011 17:33:50 (UTC Friday, 05/08/2011 05:33:50)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	4
VendorID:	058F
Product ID:	6377
Version:	1.02
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Generic USB SM Reader USB Device
Class:	DiskDrive
Device Serial Number:	920321111113&2
Mounted Devices: Last Drive Letter Mapping:	G:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d03487d-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:33:19.303
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	G:\ Friday, 05/08/2011 17:33:55 (UTC Friday, 05/08/2011 05:33:55)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:45:31 (UTC Friday, 05/08/2011 05:45:31)
Device Alert Status:	No matches

Device Number:	5
VendorID:	058F
Product ID:	6377
Version:	1.00
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	HP Photosmart C4180 USB Device
Class:	DiskDrive
Device Serial Number:	7&7949063&0&MY6C2H506V04J7&0
Mounted Devices: Last Drive Letter Mapping:	I:
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{3d034881-bf24-11e0-afdf-806e6f6e6963}
User Account NTUSER.DAT: MountPoints2:	Object was NOT used by this user
Setupapi.dev.log: First Device Connection Date/Time:	05/08/2011 17:34:02.468
Device Classes: First Connection Date/Time After Reboot:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	I:\ Friday, 05/08/2011 17:34:05 (UTC Friday, 05/08/2011 05:34:05)
USBSTOR: Last Written Date/Time Stamp:	Friday, 05/08/2011 17:36:20 (UTC Friday, 05/08/2011 05:36:20)
Device Alert Status:	No matches

Device Number:	6
VendorID:	0BC2
Product ID:	5031
Version:	Rev_210
Description:	@disk.inf,%disk_devdesc%;Disk drive
Friendly Name:	Seagate FreeAgent GoFlex USB Device
Class:	DiskDrive
Device Serial Number:	NA0ENSYX&0
Mounted Devices: Last Drive Letter Mapping or Disk Signature:	F: 30 05 54 C5
ClassGUID Number:	{4d36e967-e325-11ce-bfc1-08002be10318}
Volume GUID Number:	{078ccf70-c6f3-11e0-bc58-001fd0060cfd}
User Account NTUSER.DAT: MountPoints2:	Object was used by this user
Setupapi.dev.log: First Device Connection Date/Time:	15/08/2011 16:08:05.489
DeviceClasses: First Connection Date/Time After Reboot:	Monday, 15/08/2011 16:08:06 (UTC Monday, 15/08/2011 04:08:06)
Windows Portable Devices: Drive Name & Last Connection Date/Time:	No entry for external USB drive
USBSTOR: Last Written Date/Time Stamp:	Monday, 15/08/2011 16:08:06 (UTC Monday, 15/08/2011 04:08:06)
Device Alert Status:	ALERT: DEVICE FOUND

Completion date/time Friday, 09/12/2011 12:52:46
(UTC Thursday, 08/12/2011 23:52:46
Total Devices Found: 6
Device Alerts: 1 (Last one on device 6)
Processing time 0.66 secs

Report Notes

Device Alerts
=========

The Device Alert Status field count in this report represents the number of times identical USB device descriptor artifacts were found in the suspect evidence set and then co-located in the registry files of the analyst workstation.

This program employs a comparison analysis methodology to make a determination as to the common origin of items of evidence, by utilising registry file entries from the suspect evidence set and registry file entries of the analyst workstation after the suspect USB device had been connected to it.

Write-blocking precautions are utilised as a matter of course when the suspect USB device is connected to the analyst workstation during examination.

When USB devices are connected to a computer system, they leave unique signatures and associated entries in various system and registry files of the Windows Operating System.

Each unique "signature" can be searched for in the recovered suspect workstation registry files.

"No matches" appearing in the Alert field would indicate that the suspect USB device has not previously been connected to the suspect computer system whilst an "Alert" would indicate that the suspect USB device has previously been connected to the suspect computer by a user.

Results
=====

A blank "Mounted Devices" key field, would indicate that the drive letter was reused by another device or there was no information recorded for extraction.

An external USB hard drive and or traditional hard drive will populate the “Mounted Devices” key field with a hexadecimal disk signature value.

The "Windows Portable Devices" key field is not populated by external USB hard drive or drive letter information. Other USB devices may be recorded in this sub-key location of the SOFTWARE hive file.

This report is produced pursuant to S137(1) & S137(2) New Zealand Evidence Act 2006: "Evidence produced by machine, device, or technical process"

END REPORT

Appendix E – USBForensicReporter Verification Details

System and USB Artifact Verification Details	EnCase Forensic Software (Benchmarked Tool – Basic Bookmarks)	USBForensicReporter (Developed Prototype - HTML Report)
Suspect Computer or Host Name	T·E·S·T·-·P·C···	TEST-PC
Windows Product Name and Current Version Number	W·i·n·d·o·w·s· ·7· ·H·o·m·e· ·P·r·e·m·i·u·m··· 6·.·1···	Windows 7 Home Premium Version 6.1
User Account - NTUSER.DAT file	Test	Test
Vendor, Product and Version Identification Details	VID_0781&PID_5530 Version: Rev_1.01	VendorID: 0781 Product ID: 5530 Version: Rev_1.01
FriendlyName	S·a·n·D·i·s·k· ·C·r·u·z·e·r· ·U·S·B· ·D·e·v·i·c·e···	SanDisk Cruzer USB Device
Device - Unique Instance ID Number (Serial Number)	2005304502028AB1BCA4&0	2005304502028AB1BCA4&0
USBSTOR - ClassGUID	{4d36e967-e325-11ce-bfc1-08002be10318}	{4d36e967-e325-11ce-bfc1-08002be10318}
USBSTOR - Last Written Timestamp Values	05/08/2011 21:13:05	Friday, 05/08/2011 21:13:05
Mounted Devices Mapping	\Dos\Devices\F:	F:
Volume GUID Number (Encompassing the Device Serial Number)	{ebda7496-bf42-11e0-b600-001fd0060cfd}	{ebda7496-bf42-11e0-b600-001fd0060cfd}
DeviceClasses Information	05/08/2011 21:13:05	Friday, 05/08/2011 21:13:05
Windows Portable Devices Mapping and Last Written Timestamp	F·:·\··· 05/08/2011 21:13:08	F:\ Friday, 05/08/2011 21:13:08
NTUSER.DAT – MountPoints 2 – User Account Identification	{ebda7496-bf42-11e0-b600-001fd0060cfd}	{ebda7496-bf42-11e0-b600-001fd0060cfd}
Seupapi.dev.log – 1st Device Connection Details and Timestamp	>>> [Device Install (Hardware initiated) - USB\VID_0781&PID _5530\2005304502028AB1BCA 4] >>> Section start 2011/08/05 21:13:00.716	05/08/2011 21:13:00.716