X-Ways Software Technology AG

X-Ways Forensics/ WinHex

Integrated Computer Forensics Environment.

Data Recovery & IT Security Tool.

Hexadecimal Editor for Files, Disks & RAM.

Manual

Contents

1 Preface 1

1.1 About WinHex and X-Ways Forensics 1
1.2 Legalities 2
1.3 License Types 2
1.4 Differences between WinHex and X-Ways Forensics 3
1.5 Getting Started with X-Ways Forensics 4

2 Technical Background 5

2.1 Using a Hex Editor 5
2.2 Endian-ness 6
2.3 Integer Data Types 6
2.4 Floating-Point Data Types 7
2.5 Date Types 7
2.6 ANSI ASCII/IBM ASCII 8
2.7 Checksums, Hashes, Digests 9
2.8 Attribute Legend 10
2.9 Technical Hints 11

3 User Interface 12

3.1 Overview 12
3.2 Start Center 13
3.3 Directory Browser 14
3.3.1 General Description 14
3.3.2 Virtual Objects 16
3.3.3 Options 17
3.3.4 Filters 20
3.3.5 Columns and Column-based Filters 22
3.4 Mode Buttons 30
3.5 Status Bar 33
3.6 Data Interpreter 34
3.7 Position Manager 35
3.8 Useful Hints 35

4 Menu Reference 37

4.1 Directory Browser Context Menu 38
4.2 Data Window Context Menu 46
4.3 File Menu 47
4.4 Edit Menu 49
4.5 Search Menu 50
4.6 Navigation Menu 51
4.7 View Menu 52
4.8 Tools Menu 54
4.9 File Tools 56
4.10 Specialist Menu 57
4.11 Options Menu 60
4.12 Window Menu 60
4.13 Help Menu 61
4.14 Windows Context Menu 61

5 Forensic Features 62

5.1 Case Management 62
5.2 Multi-User Coordination For Large Cases 64

5.3 Evidence Objects 68
5.4 Case Log 69
5.5 Case Report 70
5.6 Report Tables 71
5.7 Internal Viewer 74
5.8 Registry Report 76
5.9 Simultaneous Search 78
5.10 Logical Search 79
5.11 Search Hit Lists 82
5.12 Search Term List 83
5.13 Event Lists 86
5.14 File Type Categories.txt 88
5.15 Hash Database 89
5.16 PhotoDNA 91
5.17 Time Zone Concept 92
5.18 Evidence File Containers 93
5.19 Related Items 95
5.20 External Analysis Interface 96

6 Volume Snapshots and their Refinement 97

6.1 Introduction 97
6.2 Refinement at the Volume/Sector Level 98
6.2.1 Run X-Tensions 98
6.2.2 Particularly thorough file system data structure search 98
6.2.3 File Header Signature Search 100
6.2.4 Block-wise Hashing and Matching 100
6.3 Refinement at the File Level 101
6.3.1 Hash Value Computation and Matching 101
6.3.2 File Type Verification 102
6.3.3 Extraction of Internal Metadata 103
6.3.4 Archive Exploration 104
6.3.5 E-mail Extraction 105
6.3.6 Uncovering Embedded Data 106
6.3.7 Extraction of Video Stills 108
6.3.8 Pictures Analysis and Processing 109
6.3.9 FuzZyDoc 110
6.3.10 Detection of Encryption 112
6.3.11 Indexing 112
6.4 More Information about Volume Snapshot Refinement 114
6.4.1 Interdependencies 115
6.4.2 Notes 115

7 Some Basic Concepts 116

7.1 Edit Modes 116
7.2 Scripts 117
7.3 X-Tensions API 118
7.4 WinHex API 119
7.5 Disk Editor 120
7.6 Memory Editor/Analysis 121
7.7 Template Editing 122

8 Data Recovery 123

8.1 File Recovery with the Directory Browser 123
8.2 File Recovery by Type/File Header Signature Search 123
8.3 File Type Definitions 125

8.4 Manual Data Recovery 128

9 Options 129

9.1 General Options 129
9.2 Volume Snapshot Options 135
9.3 Viewer Programs 137
9.4 Undo Options 139
9.5 Security Options 140
9.6 Search Options 141
9.7 Replace Options 145

10 Miscellaneous 146

10.1 Block 146
10.2 Modify Data 146
10.3 Conversions 147
10.4 Sector Superimposition 148
10.5 Wiping and Initializing 149
10.6 Disk Cloning 150
10.7 Images and Backups 152
10.8 Hints on Disk Cloning, Imaging, Image Restoration 156
10.9 Skeleton Images 157
10.10 Backup Manager 161
10.11 Reconstructing RAID Systems 162

Appendix A: Template Definition 165

1 Header 165
2 Body: Variable Declarations 166
3 Body: Advanced Commands 167
4 Body: Flexible Integer Variables 169

Appendix B: Script Commands 170

Appendix C: Master Boot Record 177

1 Preface

1.1 About WinHex and X-Ways Forensics

X-Ways Software Technology AG
Carl-Diem-Str. 32
32257 Bünde
Germany
Fax: +49 3212-123 2029

Web: http://www.x-ways.net
Product homepage: http://www.x-ways.net/winhex/
Ordering: http://www.x-ways.net/winhex/order.html
Support forum: http://www.winhex.net
E-mail address: mail@x-ways.com

Registered in Bad Oeynhausen (HRB 7475). CEO: Stefan Fleischmann. Board of directors (chairwoman): Dr. M. Horstmeyer.

X-Ways Software Technology AG is a stock corporation incorporated under the laws of the Federal Republic of Germany. WinHex was first released in 1995. This manual was compiled from the online help of WinHex/X-Ways Forensics v18.4, released July 2015.

Supported platforms: Windows XP, Windows 2003 Server, Windows Vista/2008 Server, Windows 7, Windows 8/Windows 2012 Server, Windows 8.1, Windows 10 Technical Preview. 32-bit and 64-bit. Standard, PE and FE.

User interface translation: Chinese by Sprite Guo. Japanese by Takao Horiuchi and Ichiro Sugiyama. French by Jérôme Broutin, revised by Bernard Leprêtre. Spanish by José María Tagarro Martí. Italian by Fabrizio Degni, updated by Michele Larese de Prata, further completed and updated by Andrea Ghirardini. Brazilian Portuguese by Heyder Lino Ferreira. Polish by ProCertiv Sp. z o.o. (LLC).

We would like to thank the state law enforcement agency of Rhineland-Palatinate for extraordinarily numerous and essential suggestions on the development of X-Ways Forensics and X-Ways Investigator.

Thanks to Dr. A. Kuiper for his method to process videos with MPlayer.

Professional users around the world include... (this list is from ~12 years ago)
U.S. and German federal law enforcement agencies, ministries such as the Australian Department of Defence, U.S. national institutes (e.g. the Oak Ridge National Laboratory in Tennessee), the Technical University of Vienna, the Technical University of Munich (Institute of Computer Science), the German Aerospace Center, the German federal bureau of aviation accident investigation, Microsoft Corp., Hewlett Packard, Toshiba Europe, Siemens AG, Siemens Business Services, Siemens VDO AG, Infineon Technologies Flash GmbH & Co. KG, Ontrack Data International Inc., Deloitte & Touche, KPMG Forensic, Ernst & Young, Ericsson, National Semiconductor, Lockheed Martin, BAE Systems, TDK Corporation, Seoul Mobile Telecom, Visa International, DePfa Deutsche Pfandbriefbank AG, Analytik Jena AG, and many other companies and scientific institutes.

1.2 Legalities

License Agreement

Acknowledgements

The MD5 message digest is copyright by RSA Data Security Inc.

The "zlib" compression library is copyright by Jean-loup Gailly and Mark Adler. Homepage: `ftp://ftp.cdrom.com/pub/infozip/zlib/zlib.html`

X-Ways Forensics contains software by Igor Pavlov, www.7-zip.com, and an Adler32 implementation by Arnaud Bouchez.

Outside In® Technology Copyright © 1991, 2014, Oracle Corp. and/or its affiliates. All rights reserved.

NEXT3® is a registered trademark of CTERA Networks.

X-Ways Forensics uses ResIL, a fork of DevIL. ResIL is governed by the LGPL (http://www.gnu.org/copyleft/lesser.html), version 2.1. The original source code can be downloaded from http://sourceforge.net/projects/resil.

X-Ways Forensics contains an unofficial build of libPFF. libPFF is governed by the LGPL (http://www.gnu.org/copyleft/lesser.html), version 3.0. The original source code can be downloaded from http://libpff.sourceforge.net/.

Windows event log (.evtx) viewing capability based on works by Andreas Schuster.

1.3 License Types

The full version of WinHex will save files larger than 200 KB, write disk sectors, edit virtual memory and show no evaluation version reminders. It will reveal its license status on start-up and in the About box. To use WinHex as a full version, you need at least one license (base license). If you are going to use WinHex on multiple machines, you will also need additional licenses.

- Personal licenses are available at a reduced price for non-commercial purposes only, in a non-business, non-institutional, and non-government environment.

- Professional licenses allow usage of the software in any environment (at home, in a company, in an organization, or in public administration). Professional licenses provide the ability to execute scripts and to use the WinHex API.

- Specialist licenses in addition to this allow to use the Specialist Tools menu section, to fully interpret exFAT, Ext2, Ext3, Ext4, Next3®, CDFS/ISO9660, and UDF media, and enable support for RAID reconstruction, Windows dynamic disks, Linux LVM2 and reverse disk cloning/imaging. Particularly useful for IT security specialists.

- Forensic licenses (i.e. licenses for X-Ways Forensics) in addition to the above allow to use the powerful case managing and report generating capabilities, the internal viewer and the separate viewer component, the gallery view, all advanced features of refined volume snapshots, all columns and filters in the directory browser, comments and report tables, plus ReiserFS, Reiser4, HFS, HFS+, UFS and XFS support. Furthermore, they allow to read and write evidence files (.e01) and **much more**! Particularly useful for computer forensic examiners. The forensic edition of WinHex is called X-Ways Forensics.

A more complete license comparison can be found online at http://www.x-ways.net/winhex/comparison.html. Please see http://www.x-ways.net/order.html on how to order your licenses.

1.4 Differences between WinHex and X-Ways Forensics

WinHex (main executable file is winhex.exe or winhex64.exe) always identifies itself as WinHex in the user interface, X-Ways Forensics (main executable file xwforensics.exe or xwforensics64.exe) as X-Ways Forensics. The shared program help and the shared manual, however, statically refer to the name “WinHex” in most cases, sometimes “X-Ways Forensics”.

WinHex and X-Ways Forensics share the same code base. X-Ways Forensics offers numerous additional forensic features over WinHex with a specialist license, but does not allow to edit disk sectors or interpreted images and lacks various functions to wipe data known from WinHex. In X-Ways Forensics, disks, interpreted image files, virtual memory, and physical RAM are strictly opened in view mode (read-only) only, to enforce forensic procedures, where no evidence must be altered in the slightest. This strict write protection of X-Ways Forensics ensures that no original evidence can possibly be altered accidentally, which can be a crucial aspect in court proceedings.

Only when not bound by strict forensic procedures and/or when in need to work more aggressively on disks or images (e.g. you have to repair a boot sector or wipe classified or unrelated data), then a user of X-Ways Forensics would run WinHex instead. With WinHex you can edit disk sectors and wipe entire hard disks, free space, slack space, selected files, selected disk areas etc.

Since v17.1, users of X-Ways Forensics may simply copy their xwforensics.exe executable file and name the copy winhex.exe (or for the 64-bit edition copy their xwforensics64.exe executable file and name the copy winhex64.exe) to get WinHex. Or they may use the setup program, which creates hard links named like that in the target directory. Or they can create hard links themselves (which is much cooler than ordinary copying). If the program is executed as *winhex*.exe, it will identify itself as WinHex everywhere (in the user interface, case report, case log, image descriptions, and all screenshots) and act/behave like WinHex. That version is the best of both worlds, with the full forensics feature set of X-Ways Forensics plus the sector editing and data wiping capability of WinHex in one.

The WinHex API can only be used in conjunction with WinHex.

1.5 Getting Started with X-Ways Forensics

For the latest download instructions, if your update maintenance is current, you can check your license status here. For more information about the installation of WinHex and X-Ways Forensics please see this web page.

Extract the files in the X-Ways Forensics download to a directory of your choice. An installation with the setup program is not necessary. The program is portable and can also be started directly from a USB stick on other computers, e.g. live systems that you would like to examine. Also download the viewer component (which is not included in the standard download as it is updated much more rarely). Use the 64-bit edition of the viewer component for the 64-bit edition of X-Ways Forensics. By default, the viewer component is expected in the subdirectory \viewer (32 bit) or \x64\viewer (64 bit). Please be advised that the viewer component creates files in the profiles of the user who is currently logged on, unlike X-Ways Forensics, so if you wish to avoid to create files on a live system that you examine, don't let X-Ways Forensics use the viewer component. You may also wish to download MPlayer if you intend to have X-Ways Forensics produce stills from videos to see them in the gallery. Newer releases can always be extracted into the existing directory of an earlier release. You may continue to use WinHex.cfg configuration files from earlier releases in later releases (but never the other way around).

Here are some instructions to help you get started and find some important features: Create a case, add an evidence object (such as your own C: drive or hard disk 0, or an image file). In the directory tree, you may use a right click to list the contents of a directory in the directory browser including all its subdirectories. For example, if you right-click the root directory of a volume, you will get a listing of all files in the entire volume. At the same time you can use a dynamic filter to focus on files based with certain filenames, of a certain file type, size, or with certain timestamps, etc. via Options | Directory Browser.

The powerful logical search functionality can be found in Search | Simultaneous Search. More interesting functions in X-Ways Forensics can be found in the context menu of the directory browser (e.g. the ability to copy files off an image) and in the Specialist menu, in particular “Refine Volume Snapshot”). The latter allows you to further process files automatically, e.g. explore zip archives, extract e-mail messages and attachments, check pictures for the amount of skin tones, check documents for encryption, etc.

There are a thousand different purposes for which X-Ways Forensics can be used, so in our opinion step-by-step instructions (click here first, then there, then look here) are not the right way to explain the software. This program help/user manual is rather meant to accurately describe all the available functionality and let you creatively combine different commands to achieve a certain goal. It is still the user who has to do the thinking, know what he/she is doing and how to interpret findings.

The 64-bit edition is recommended especially in situations where the 32-bit memory address space may be insufficient, when dealing with disks or images that contain many millions of files, or when dealing with many millions of search hits, provided that you have plenty of physical RAM installed. Certain operations that are computationally intensive (e.g. hashing or encrypting) may also be faster in the 64-bit edition.

2 Technical Background

2.1 Using a Hex Editor

A hex editor is capable of completely displaying the contents of each file type. Unlike a text editor, a hex editor even displays control codes (e.g. linefeed and carriage-return characters) and executable code, using a two-digit number based on the hexadecimal system.

Consider one byte to be a sequence of 8 bits. Each bit is either 0 or 1, it assumes one of two possible states. Therefore one byte can have one of $2•2•2•2•2•2•2•2 = 2^8 = 256$ different values. Since 256 is the square of 16, a byte value can be defined by a two-digit number based on the hexadecimal system, where each digit represents a tetrade or nibble of a byte, i.e. 4 bits. The sixteen digits used in the hexadecimal system are 0-9, A-F.

You can change the value of a byte by changing these digits in the hexadecimal mode. It is also possible to enter the character that is assigned to a certain byte value by a character set (cf. Entering Characters). All kinds of characters are allowed (e.g. letters and punctuation marks). Example: A byte whose decimal value is 65 is displayed as 41 in hexadecimal notation (4•16+1=65) and as the letter A in text mode. The ASCII character set defines the capital letter A to have the decimal value of 65.

When editing files of a certain type (for instance executable files), it is essential not to change the file *size*. Moving the addresses of executable code and included data results in severely damaging such files. Please note that changing the contents of a file generally may be the reason for the corresponding application to behave anomalously. It is quite safe to edit text passages in a file. At any rate, it is recommendable to create backup files before editing.

The command “Combined Search” was especially designed for editing files created by computer games to save the game state. If you know the value of a variable in two of such files, you can find out the offset, i.e. the position, at which this data is saved. Example: If two files hold the

information that you have 5 resp. 7 points/lives/..., search simultaneously for the hex value 05 in the first and 07 in the second file.

2.2 Endian-ness

Microprocessors differ in the position of the least significant byte: Intel®, MIPS®, National Semiconductor, and VAX processors have the least significant byte first. A multi-byte value is stored in memory from the lowest byte (the "little end") to the highest byte. For example, the hexadecimal number 12345678 is stored as 78 56 34 12. This is called the *little-endian* format.

Motorola and Sparc processors have the least significant byte last. A multi-byte value is stored in memory from the highest byte (the "big end") to the lowest byte. For example, the hexadecimal number 12345678 is stored as 12 34 56 78. This is called the *big-endian* format.

2.3 Integer Data Types

Format/Type	Range	Example
signed 8 bit	-128...127	FF = -1
unsigned 8 bit	0...255	FF = 255
signed 16 bit	-32,768...32,767	00 80 = -32,768
unsigned 16 bit	0...65,535	00 80 = 32,768
signed 24 bit	-8,388,608...8,388,607	00 00 80 = -8,388,608
unsigned 24 bit	0...16,777,215	00 00 80 = 8,388,608
signed 32 bit	-2,147,483,648...2,147,483,647	00 00 00 80 = -2,147,483,648
unsigned 32 bit	0...4,294,967,295	00 00 00 80 = 2,147,483,648
signed 64 bit	-2^{63} ($\approx -9 \cdot 10^{18}$)...$2^{63}-1$ ($\approx 9 \cdot 10^{18}$)	00 00 00 00 00 00 00 80 = -2^{63}

Unless stated otherwise, multi-byte numbers are stored in little-endian format, meaning that the first byte of a number is the least significant and the last byte is the most significant. This is the common format for computers running Microsoft Windows. Following the little-endian paradigm, the hexadecimal values 10 27 can be interpreted as the hexadecimal number 2710 (decimal: 10,000).

The Data Interpreter is capable of interpreting data as all of the aforementioned integer types, plus unsigned 48-bit integers.

2.4 Floating-Point Data Types

Type	Range	Precision [Digits]	Bytes
Float (Single)	$\pm 1.5^{-45}...3.4^{38}$	7-8	4
Real	$\pm 2.9^{-39}...1.7^{38}$	11-12	6
Double (Double)	$\pm 5.0^{-324}...1.7^{308}$	15-16	8
Long Double (Extended)	$\pm 3.4^{-4932}...1.1^{4932}$	19-20	10

The type names originate from the C programming language. The corresponding Pascal names are specified in brackets. The Real type exists only in Pascal. The Data Interpreter is capable of translating hex values in an editor window into floating-point numbers of all four types and vice-versa.

In the computer, a floating-point number F is represented by a mantissa M and an exponent E, where $M \times 2^E = F$. Both M and E are signed integer values themselves. The four data types differ in their value ranges (i.e. the number of bits reserved for the exponent) and in their precision (i.e. the number of bits reserved for the mantissa).

On Intel®-based systems, calculations upon floating-point numbers are carried out by a math coprocessor while the main processor waits. The Intel® 80x87 uses 80-bit precision for calculations, whereas RISC processors often use 64-bit precision.

2.5 Date Types

The following date formats are supported by the Data Interpreter:

- **MS-DOS Date & Time (4 bytes)**

The lower word determines the time, the upper word the date. Used by several DOS function calls, by the FAT file systems and many system utilities such as file archivers.

Bits	**Contents**
0-4	Second divided by 2
5-10	Minute (0-59)
11-15	Hour (0-23 on a 24-hour clock)
16-20	Day of the month (1-31)
21-24	Month (1 = January, 2 = February, etc.)
25-31	Year offset from 1980

- **Win32 FILETIME (8 bytes)**

The FILETIME structure is a 64-bit integer value representing the number of 100-nanosecond intervals since January 1, 1601. Used by the Win32 API.

- **OLE 2.0 Date & Time (8 bytes)**

A floating-point value (more exactly: a double) whose integral part determines the number of days passed since December 30, 1899. The fractional part is interpreted as the day time (e.g. 1/4 = 6:00 a.m.). This is the OLE 2.0 standard date type, e.g. it is used by MS Excel. ICQ 7.0 uses big-endian OLE 2.0 timestamps in chat messages

- **ANSI SQL Date & Time (8 bytes)**

Two consecutive 32-bit integer values. The first one determines the number of days since November 17, 1858. The second one is the number of 100-microsecond intervals since midnight. This is the ANSI SQL standard and used in many databases (e.g. InterBase 6.0).

- **UNIX, C, FORTRAN Date & Time (4 bytes)**

A 32-bit integer value that determines the number of seconds since January 1, 1970. This data type was used in UNIX, by C and C++ ("time_t"), and by FORTRAN programs since the 80's. Sporadically defined as the number of *minutes* since January 1, 1970. The Data Interpreter options let you switch between both sub-types.

- **Macintosh HFS+ Date & Time (4 bytes)**

A 32-bit integer value that determines the number of seconds since January 1, 1904 GMT (HFS: local time). The maximum representable date is February 6, 2040 at 06:28:15 GMT. The date values do not account for leap seconds. They do include a leap day in every year that is evenly divisible by 4.

- **Java Date & Time (8 bytes)**

A 64-bit integer value that specifies the number of milliseconds since January 1, 1970. Usually stored in big endian, which is the typical byte order in Java, but in little endian in BlackBerry memory.

- **Mac Absolute Time, a.k.a. Mac epoch time (4 bytes)**

A 32-bit integer value that determines the number of seconds since January 1, 2001.

2.6 ANSI ASCII/IBM ASCII

ANSI ASCII is the name utilized in WinHex for an extension of the ASCII character set as used in non-Unicode Windows applications. It was named ANSI by Microsoft after the American National Standards Institute, but not defined by that institute. Several different regional variants exist, one of which is active in Windows, typically code page 1252 in countries where a Western European language is spoken. MS-DOS and Windows command prompt windows use what is called the IBM ASCII character set in WinHex (also called OEM or DOS character set elsewhere). All of these 8-bit extensions of the 7-bit ASCII character sets differ in the characters

with values greater than 127. If for example if you store plain text file with Windows Notepad in ANSI encoding and later view it with the type command in a command prompt window, special characters such as German umlauts will not be displayed correctly. Some of the regional ANSI code pages are double-byte code pages, i.e. use even 2 bytes for some characters instead of just 1 per character.

Select the character set for the text column in the View menu. Use the "Convert" command of the Edit menu to convert text files from one character set to the other.

The first 32 ASCII values do not define printable characters, but control codes:

Hex	Control Code	Hex	Control Code
00	Null	10	Data Link Escape
01	Start of Header	11	Device Control 1
02	Start of Text	12	Device Control 2
03	End of Text	13	Device Control 3
04	End of Transmission	14	Device Control 4
05	Enquiry	15	Negative Acknowledge
06	Acknowledge	16	Synchronous Idle
07	Bell	17	End of Transmission Block
08	Backspace	18	Cancel
09	Horizontal Tab	19	End of Medium
0A	Line Feed	1A	Substitute
0B	Vertical Tab	1B	Escape
0C	Form Feed	1C	File Separator
0D	Carriage Return	1D	Group Separator
0E	Shift Out	1E	Record Separator
0F	Shift In	1F	Unit Separator

2.7 Checksums, Hashes, Digests

A checksum is a characteristic number used for verification of data authenticity. Two files with equal checksums are highly likely to be equal themselves (byte by byte). Calculating and comparing the checksums of a file *before* and *after* a possibly inaccurate transmission may reveal transmission errors. An unaffected checksum indicates that the files are (in all likelihood) still identical. However, a file can be manipulated on purpose in such a way that its checksum remains unaffected. Digests are used instead of checksums in such a case, where malicious (i.e. not mere random) modifications to the original data are to be detected.

In WinHex, checksums can be calculated for example with a command in the Tools Menu.

The standard checksum is simply the sum of all bytes in a file, calculated on an 8-bit, 16-bit, 32-bit, or 64-bit accumulator. The CRC (cyclic redundancy code) is based on more sophisticated algorithms, which are safer.

Example: If a transmission alters two bytes of a file in such a way that the modifications are countervailing (for instance byte one +1, byte two -1), the standard checksum remains unaffected, whereas the CRC changes.

A so-called digest is, similar to a checksum, a characteristic number used for verification of data authenticity. But digests are more than that: digests are *strong one-way hash codes*.

It is computationally feasible to manipulate any data in such a way that its checksum remains unaffected. Verifying the checksum in such a case would lead to the assumption that the data has not been changed, although it has. Therefore, digests are used instead of checksums if malicious (i.e. not mere random) modifications to the original data are to be detected. It is computationally infeasible to find any data that corresponds to a given digest. It is even computationally infeasible to find two pieces of data that correspond to the same digest.

Of course, random modifications, e.g. caused by an inaccurate transmission, can also be detected when using digests, but checksums are sufficient and serve better for this purpose, because they can be calculated much faster.

WinHex can compute the following digests: MD4, MD5, SHA-1, SHA-256, RipeMD-128, RipeMD-160, Tiger128, Tiger160, Tiger192 as well as TTH (Tiger Tree Hash) and ed2k (specialist and forensic licenses only).

2.8 Attribute Legend

A: to be archived
R: read-only
H: hidden
S: system
X: not indexed
P: NTFS reparse point
O: offline
T: temporary
I: has object ID
C: compressed at filesystem level
c: compressed in archive
E: encrypted at filesystem level
e: encrypted in archive
e!: file type specific encryption/DRM
e?: high entropy, possibly fully encrypted
(Res): HFS+ resouce
($EFS): NTFS encryption metadata
(INDX): NTFS non-directory index attribute
(ADS): NTFS alternate data stream
(SC): found in a volume shadow copy
(SUID): Set User ID
(SGID): Set Group ID

File mode:
l=symbolic link

c=character device
b= block device
s=socket
p=pipe

Permissions:
owner read/write/execute
group read/write/execute
other read/write/execute

2.9 Technical Hints

- Technical specifications

Supported disk and file size:	at least 120 TB
Supported file size in volume snapshots:	120 TB-1
Maximum number of sectors generally:	2^{40}-1
Maximum number of clusters generally:	2^{32}-1
File system supported for volumes > 2^{32} sectors:	NTFS, Ext*, XFS, Reiser*
File system supported for volumes > 2^{32} clusters:	NTFS, Ext4, XFS
Maximum number of simultaneously open interpreted disk images	100
Maximum number of simultaneously open partitions and interpreted volume images	256
Maximum number of data windows:	1000
Maximum number of simultaneous program instances:	99
Maximum number of reversible keyboard inputs:	65535
Encryption depth:	128-256 bit
Offset presentation:	hexadecimal/decimal

- In most cases, the progress display shows the completed percentage of an operation. However, during search and replace operations it indicates the relative position in the current file or disk.

- Keys you specify for encryption/decryption are not saved on the hard disk. Provided that the corresponding security option is enabled, the key is stored in an encrypted state within the RAM, as long as WinHex is running.

- Search and replace operations generally run fastest with case sensitivity switched on and without wildcards enabled.

- When searching with the option “count occurrences” activated or when replacing without prompting, for a search algorithm there are generally two ways to behave when an occurrence has been found, which in some cases may have different results. This is explained by the following example:

  The letters *ana* are searched in the word “banana”. The first occurrence has already been found at the second character.
  1st alternative: The algorithm continues the search at the third character. So *ana* is found

again at the fourth character.

2nd alternative: The three letters *ana* found in the word “banana” are skipped. The remaining letters *na* do not contain *ana* any more.

WinHex is programmed in the second manner, because this delivers the more reasonable results when counting or replacing occurrences. However, if you continue a search using the **F3** key or you choose the replace option “prompt when found”, the algorithm follows the first paradigm.

Special Performance Enhancements

File header signature searches, block-wise hash matching, FILE record searches, searches for lost partitions, and physical simultaneous searches are now sparse-aware operations when dealing with compressed and sparse .e01 evidence files. That means that areas that on the original hard disk were never written and zeroed out or areas that had been wiped on the original hard disk or consciously omitted areas in cleansed images are skipped and almost require no time, because their data neither has to be read nor decompressed nor further processed (searched / hashed / matched against the block hash database).

Sparse-awareness is active guaranteed for .e01 evidence files that were created by X-Ways Forensics and X-Ways Imager 16.1 and later (also possibly for images created by 3rd party software, depending on the settings and the internal layout). Operations are not sparse-aware on images of Windows dynamic disks, images of LVM2 disks, and on reconstructed RAIDs based on .e01 evidence files.

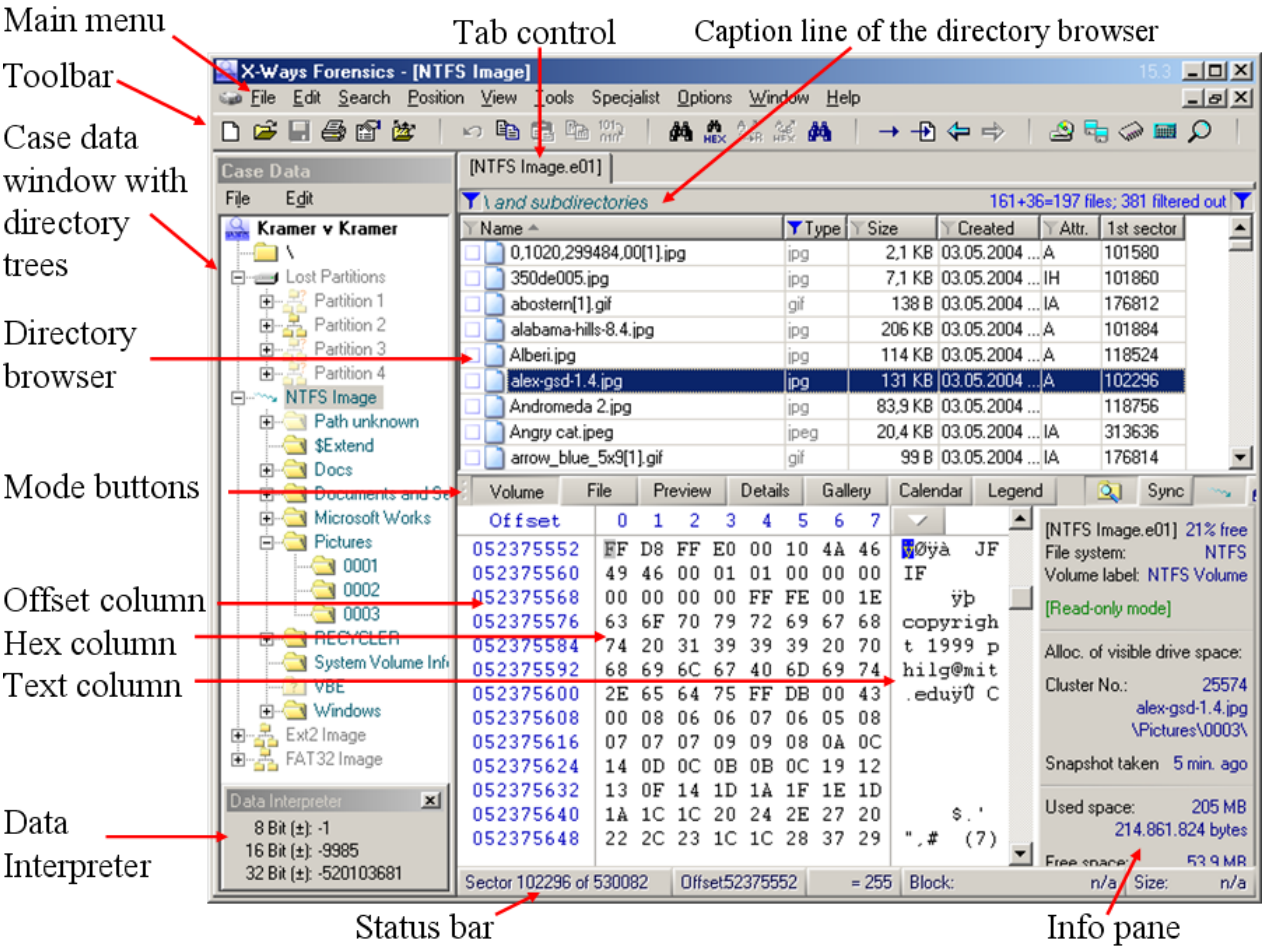
Logical searches in files stored in an NTFS file system are also sparse-aware at the .e01 evidence file level, and generally logical searches in virtual "Free space" files.

Logical searches in NTFS, Ext*, XFS and UFS file systems are now sparse-aware at the file system level. That means no time is wasted on large sparse areas within sparse files. Those areas are ignored, regardless of whether the evidence object is an .e01 evidence file, raw image, RAID, or actual disk.

3 User Interface

3.1 Overview

To familiarize yourself with the names of the various elements of the user interface, please refer to this screenshot:



3.2 Start Center

The so-called Start Center is a dialog window that is optionally displayed at startup and is meant as a simplified control panel for beginning your work. It allows to quickly open files, disks, memory modules, and folders as well as up to 255 recently edited documents (16 by default, left-hand list). These may be files, folders, logical drives or physical disks. When opened again, WinHex restores the last cursor position, the scrolling position, and the block (if defined) of each document, unless the corresponding option is disabled.

From the Start Center you are also able to access *projects* and *cases* (right-hand top list). A project consists of one or more documents to edit (files or disks). It remembers the editing positions, the window sizes and positions and some display options. By saving a window arrangement as a project you can continue to work in several documents right where you left them, with a single click only. This is especially useful for recurring tasks. When you load a project, all currently opened windows are automatically closed first.

Besides, WinHex automatically saves the window arrangement from the end of a WinHex session as a project, and can re-create it next time at startup. Each project is stored in a .prj file. It can be deleted or renamed right within the Start Center (context menu or DELETE/F2 key).

Last not least, the Start Center is the place where to manage *scripts*. You may check, edit, create, rename, and delete scripts using the context menu. To execute a script, double-click it or single-click it and click the OK button.

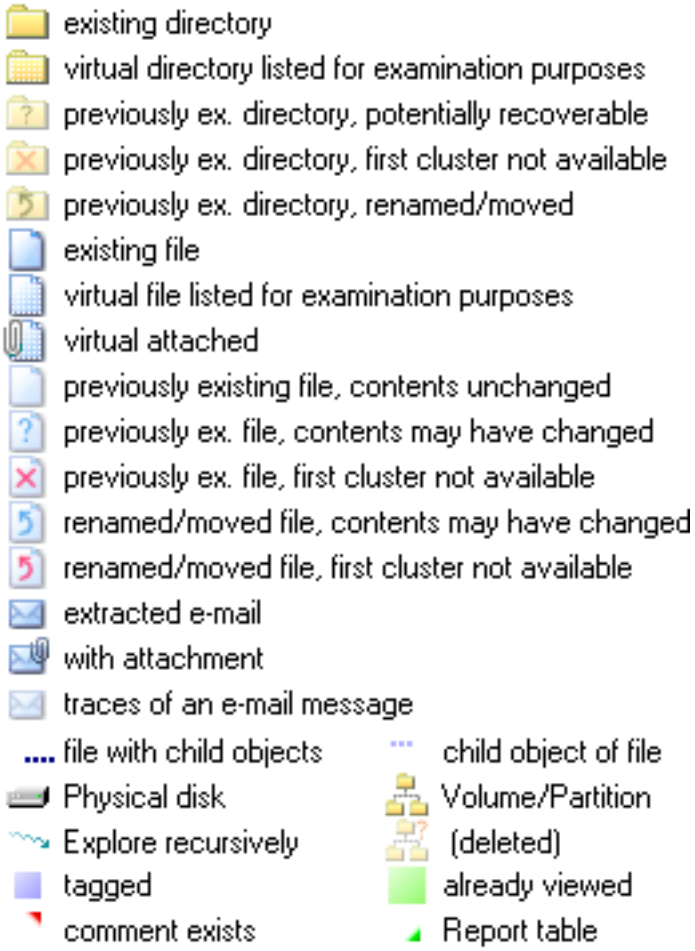
3.3 Directory Browser

3.3.1 General Description

The perhaps most essential user interface element in WinHex and X-Ways Forensics is the so-called *directory browser*, which resembles the Windows Explorer's right-hand list. Its main task is to display (and interact with) the volume snapshot. Complete functionality is available only with a forensic license. By default, the directory browser lists directories first, then files. Compressed files are displayed in blue, encrypted files in green. Right-clicking any item in the directory browser brings up a context menu with commands for opening a file or directory, exploring a directory, locating the beginning of a file or directory on the disk, locating the corresponding directory entry (FAT) or file record (NTFS), listing the allocated clusters in a separate window, etc.

When navigating from one directory to another, exploring files with child objects (e.g. e-mail messages that have attachments), navigating to the parent of a child object, activating or deactivating filters, trying different sort criteria etc., please note that you can easily return to a previous view using the Back command in the Position menu or the Back button in the toolbar.

The **icons** are explained in the legend directly in the program (forensic license only). Deleted files and directories are represented in the directory browser with lighter icons. Icons with a blue question mark indicate that the original file or directory contents may be still available. Deleted objects that WinHex knows are no longer accessible (either because their first cluster has been reallocated, because it is unknown, or because they have a size of 0 bytes) have icons crossed out in red. Icons with an arrow on FAT volumes (only with a specialist or forensic license) and (after refining the volume snapshot) NTFS volumes show renamed and moved files with their original name/in their former directory. On Reiser4 these are moved files with their current name in their former directory. A blue arrow indicates that contents for a file are available (though these are not specifically the contents from before the file was renamed or moved). A red arrow indicates that no contents are available.



In the caption line of the directory browser you see on the left the explored path (in case of recursive exploration in italics and turquoise color). When clicking any component of the current path, this will now navigate directly to that directory (or file with child object) whose name you clicked. On the right you see the number of listed files and directories (typically separate figures for existing objects + previously existing objects + virtual objects). Also, the number of listed tagged files is indicated, if any are tagged. The number of active filters is displayed as well, next to the blue filter symbol on the left. Column-based and column-independent active filters are counted separately. Useful because there might be column-based filters active for columns that are not currently visible in the directory browser, and that column-independent filters are active may be otherwise apparent only when checking in the directory browser options dialog.

The directory browser can **sort** files and directories in ascending or descending order, and still reveals the two previous sort criteria with a lighter arrow. For example, if you first click the filename column and then the filename extension column, files with the same extension will internally still be sorted by name.

In order to undefine the secondary and tertiary sort criteria, hold the Shift key when clicking on the column header to determine the primary sort criterion. Internally, this selects the internal ID as the secondary sort criterion. This is to ensure that the order of items with identical data for the primary sort criterion is still well defined and reproducible after having sorted by other sort criteria in the meantime.

The column that functions as the primary sort criterion is also the target of “jump as you type”. That is, you can type the first character or first few characters of the entry that you are looking for when the directory browser has the focus to automatically navigate and select the first or next matching item in the list, starting from the current position. For example, if the directory browser is sorted by the Type column, type “z” if you wish to find the first zip file in the list. If however there is another file listed with a type starting with “z”, one that precedes “zip” alphabetically, for example “zac”, then type the next character (before the feature times out and forgets the “z” that you have already entered), in this case “i”, until you find what you are looking for or nothing happens any more (if there is no matching item). Matching occurs in a cycle. That means even if the current position shows a zip file, you can type any preceding letter to jump to the first matching item from the top again, for example “d” for .docx. If you are looking for .docx files, but find a large group of .doc files, then you need to type all four characters of docx, because only the “x” distinguishes docx from doc.

3.3.2 Virtual Objects

When orphaned objects are found, e.g. files that have been deleted and whose original path is unknown, they are listed in a special virtual directory “Path unknown”. With a specialist or forensic license, there are virtual files in the root directory that allow you to conveniently address special areas in a volume:

File system areas: Reserved sectors and/or clusters that are claimed by the file system itself for internal purposes.

Free space: Clusters marked by the file system as not in use. Depends on the volume snapshot options.

Idle space: Areas in a volume of which WinHex does not know what they are used for, including in particular clusters marked by the file system as in use, whose exact allocation however could not be determined. This can be the case if the file system lost track of them, i.e. forgot that these cluster are actually available for re-allocation. Usually there is no idle space. The size of idle space and the number of the first idle cluster are only determined when needed (e.g. when you click the "Idle space" file for the first time), as depending on the number of cluster this is a potentially time-consuming operation.

Volume slack: Sectors at the end of the partition that are unused by the file system because they do not add to another cluster.

Indirect blocks (Ext2, Ext3, UFS): Special blocks that contain block numbers. Not part of "File system areas".

Unnoted attribute clusters (NTFS): Clusters that contain non-resident attributes that have not been individually processed by X-Ways Forensics. Not part of "File system areas".

.journal (ReiserFS): Blocks that form the fixed journalling area. On Ext3 and HFS+, this is not

considered a virtual file because it is defined by the file system itself in dedicated records.

3.3.3 Options

- **Grouping files and directories** in the directory browser is optional. X-Ways Forensics remembers the sort criteria and this option separately 1) for the normal directory browser of a volume, 2) for the normal directory browser of a partitioned disk, 3) for search hit lists and 4) for event lists.

- **Grouping existing and deleted items** in the directory browser is optional. There are two possibilities how to use this feature. Either previously existing files that potentially recoverable (question mark icon) and known unrecoverable (red X icon) are internally grouped as well (so that in total there will be three groups) or not (only 2 groups). A small symbol with either one or two horizontal dividers indicates whether the list is split up into two or three groups, also in the header of the column that is the primary sort criterion, as a small reminder that when scrolling in the directory browser and watching out for a certain file for example based on its name, you need to check in every group, because the sorting takes place within each group and does not span the groups.

- A ".." item can be optionally listed at the top of the directory browser when navigating within a volume from one directory to another. If displayed, it is frozen at the top and does not scroll along with all the other items. It shows all the information on the directory that it represents (the one that you would navigate to if you double-click it), just like with all the other items in the directory browser. A "." item is also displayed optionally, representing the currently explored directory. Useful if for example you wish to see certain metadata (e.g. timestamps) of the parent object at the same time as metadata of its child objects. And if the . or .. item is a file and you select it, then you can see that particular file in File, Preview or Details mode. And it is represented in Gallery mode.

- **Double-clicking** a directory will **explore** it. Double-clicking an ordinary file will **view** it. This option controls whether files with child objects will be typically viewed or explored on a double-click. If the checkbox is half-checked, you will be prompted.

- Files can optionally be **opened and searched** including their **slack**. The middle state of this checkbox makes a difference only for logical searches (cf. that topic).

- Listing sub**dir**ectorie**s** when **exploring recursively** is optional. They may be needed if you are interested in their names or timestamps, but they may distract you when you are merely interested in viewing files.

- The **selection statistics** are displayed below the directory browser (with a forensic license only). If computed in a **recursive** way, they reveal how many subdirectories, files and how much data are contained in a directory when you select it in the directory browser, except if you have explored recursively already, taking any active filters into account. If this option is not enabled, only the statistics tell you about the direct selection in the directory browser only,

not about the child objects that may indirectly be selected via selected directories. If this option is half selected, the statistics take child objects of directories into account, but not child objects of files.

- **Tagging** or **excluding** items in the directory browser can occur **recursively** or non-recursively. Non-recursively means that tagging/untagging/excluding/including a file or directory in the directory browser has no effect on parent or child objects or parent directories or subdirectories. Useful for example if all child objects of a file should be processed in volume snapshot refinement or searched, but not the parent object. If it works recursively, then it is not possible to have an untagged parent object whose child objects are all tagged. If the recursive tagging option is in its middle state, that means that child objects still inherit the tagged state from their parent at the moment when they are newly added to the volume snapshot, e.g. when you extract e-mail and attachment from a tagged e-mail archive. Whether tagging and excluding work recursively or not can also be controlled by holding the Shift key. Tagging or untagging recursively can be *very* slow in large volume snapshots.

- An option exists to **show** the **file type ranks** in the Type status column, which also causes sorting by that column to **sort** by those ranks. Ranks are defined in the File Type Categories.txt file.

- **Advanced sorting**: Takes 4 to 6 times more time than the highly optimized standard Unicode sorting (noticeable when sorting millions of files), but has several useful settings and characteristics:
  - Language-specific character equivalence rules (treat ß like ss, treat é similar to e, ü similar to u etc.)
  - Linguistically improved case insensitivity
  - Special treatment of hyphens and apostrophes (they are treated differently from other non-alphanumeric characters to ensure that words such as "coop" and "co-op" stay together in a sorted list).
  - Treat decimal digits as numbers, e.g. sort "2" before "10" (not useful for hexadecimal notation, available under Windows 7 and later only)
  - Treat half-width and full-width characters the same (full-width characters are sometimes used by East Asians when writing English language letters)
  - Ignore kana type (treat corresponding Japanese hiragana and katakana characters the same)

  Advanced sorting depends on the regional settings of the currently logged on user. For example, if regional settings of a Nordic country are active, Å comes after Z, as defined in the alphabets of that region, otherwise near A, as perhaps expected by non-locals. Advanced sorting rules are also applied when sorting the search hits by the Search Hit column.

  There is an option to sort search hits by their data and context instead of just by the search terms to which they belong. Helpful for keyword searches (not technical, e.g. hex value, searches). Indeed slower since the data and context of all search hits to sort have to be read and converted to a comparable code page. Sorting by the data in search hits helps for GREP searches. It makes a difference only for GREP expressions that match variable data because for constant search terms the search terms and the data in their corresponding search hits are identical. For example, after searching for e-mail addresses with the expression [a-zA-Z0-9_\-\+\.]{1,20}@[a-zA-Z0-9\-\.]{2,20}\.[a-zA-Z]{2,7}, sorting by the data allows you to quickly

identify and visually skip groups of identical e-mail addresses or see similar e-mail addresses (starting with the same characters) next to each other. Continuing sorting by the text that follows the actual search hit if the search hit data is the same will show identical or similar text passages next to each other and allow you to more quickly review the search hit list. You can specify how many characters of data and context to take into account for sorting. The more characters, the more memory is needed for sorting, which can make a difference when listing a huge number of search hits.

Optionally, the names of directories and file with child objects can be included when **sorting** by **path** (full path sorting). The effect is that the child **obj**ects will be listed directly after their respective parents (e.g. e-mail attachments after their containing **parent** e-mail messages).

- Optionally, after **start-up**, the directory browser can be not **sort**ed at all, for performance reasons. That means the program will forget the last sort criteria in use last time. If selected, there will now also be no sorting when turning off all filters with a single mouse click, to avoid longer delays when suddenly all files are listed again recursively.

- **Directory browser settings** (in particular column width, filter settings and sort orders) can be optionally **stored in cases** and reactivated when loading cases (if stored by a compatible version).

- **Dynamic e-mail and timestamp columns** lets X-Ways Forensics decide whether to include the columns Sender and Recipient in the directory browser. They will be included if at least one extracted e-mail message is in the visible portion of the directory browser, otherwise not. Helpful because that leaves more room for other columns when the columns exclusively filled for extracted e-mail messages are not needed. The columns with alternative timestamp can also be shown dynamically, i.e. only when items that have such timestamps in the volume snapshot are displayed in the visible portion of the directory browser.

- The 1st sector column can optionally show physical start sector numbers for files in partitions (counted from the start of the physical disk or disk image) instead of logical start sector numbers, if the partition was opened from within the physical disk/disk image. In that case the column label contains a P in a circle (P for physical). Only for ordinary partitions, not Windows dynamic volumes or LVM2 volumes.

- **SHA-1 hashes** can always be **displayed in Base32** notation in the directory browser, as common in P2P programs.

- **Conditional cell** background **coloring** helps to draw your attention to items of interest without having to filter out all non-matching items. Matching items are found through a substring search in the cell contents of a selected column. Substring expressions may be up to 15 characters long. You may use an asterisk to match anything except blank cells. If a match is detected in a cell, either only the background of that particular cell can be colored (called "cell-targeted coloring") or the entire line. To color an entire column, regardless of the cell contents, activate cell-targeted coloring for that column and specify an empty condition string, i.e. no condition at all. If a cell meets multiple cell-targeted conditions or multiple line-targeted conditions, only the first condition of each group will be applied. If different

conditions apply to the same cell (one cell-targeted and one line-target color), that cell will be shown in a mix of both colors. For line-targeted coloring, only the first 255 characters in the respective cell are guaranteed to be searched.

Conditions cannot be defined for search hit specific columns, but for event specific columns. That can prove useful when trying to identify patterns in events. For example, you could color all events of type "Program started" in red and log-in events in yellow and see more easily how far apart from each other they are. Conditional cell background coloring is case-specific if "Store directory browser settings in cases" is selected. The color settings are also stored in a file named "Conditional Coloring.cfg", and they are stored in and loaded from .settings files along with other directory browser settings. Up to 255 conditions may be defined.

Columns

Various columns are available in the directory browser. They are all optional. They are displayed if they have a non-zero column width in pixels, or hidden if their width is zero. You can toggle column visibility purely with the mouse if you like, by clicking the column label in the dialog window.

It is possible to redefine the *order* of the columns in the directory browser. This will also change the order of the fields in the case report (i.e. in report tables), on print cover pages, in exported file listings, and the Export/Copy log. You can select a column for relocation by clicking its radio button. Then use the vertical scrollbar that appears at the top. You can reset the column order to the default one by *right*-clicking that scrollbar.

3.3.4 Filters

The following can be dynamically filtered out (by choosing to not list it):

- Existing files. Useful if you are merely interested in previously existing files (which could reside in existing directories).
- Previously existing files and directories.
- Tagged files and directories.
- Half tagged files and directories (that contain at least 1 tagged and at least 1 untagged file).
- Untagged files and directories.
- Files that are marked as already viewed.
- Files that are not marked as already viewed.
- Excluded files and directories (marked as excluded in the volume snapshot).
- Files and directories that are *not* excluded.

You may also activate filters based on criteria such as filenames, file type categories, attributes, or hash set. Whenever an active filter actually filters out files or directories in the directory browser, this is flagged with a blue filter icon in the directory browser's header line, and you will be informed of how many items exactly have been omitted from the list. You also have the option, by clicking the icons for "open file"/"save file" on the right-hand side of the caption line

of the directory browser, to store filter and sort settings in a separate file and load them again at any time. Such files are given the extension ".settings". Note that it is not guaranteed that different versions of the software can load each other's settings.

Below the filter options in the lower left corner you will find a button in this dialog box that allows to undo the exclusion of all files and directories in the volume snapshot of the evidence object in the active data window. To selectively include files, make sure they are not filtered out. Then you can include them with a context menu command after selecting them.
There is another button that allows to totally remove excluded items from the volume snapshot if irrelevant/not needed, in particular meaningless garbage files found via a file header signature search. This will render the volume snapshot smaller, i.e. more efficient to handle, and save main memory. Useful also if you would like X-Ways Forensics to find certain files once again via a file header signature search, but for example list them with a different default file size if the originally specified default file size proved inadequate. The removal operation is faster if you delete seach hits prior to executing it. As part of the removal, internal IDs are shuffled, so they do not indicate any more the order in which items were added to the volume snapshot. Excluded items that have non-excluded child objects are not removed. It is highly recommended to work with a copy of your case when using this functionality, e.g. produced with the Save As command.

Whenever one or more filters are active that actually filter out items in the currently displayed directory browser, there are two blue filter symbols in the directory browser's caption line. They point out that your current view is incomplete because of active files, and they also allow you to deactivate *all* filters with a single mouse click, to ensure you are not missing any file when you no longer want the filter. You can activate or deactivate column-based filters individually with a single mouse click on the column header's filter symbol when holding the Shift key. The options of the respective filter remain unchanged in this case.

The filters have been given some "intelligence" when navigating from a parent file to a child file or vice-versa, so that the filters "know" when it's a good time to be turned off.
For example:
- If you are using a filter to focus on all extracted e-mail messages recursively, and then you double-click an individual e-mail message to have a look at its attachments in the directory browser, the filter is automatically deactivated, so that you can actually see these attachments. A simple click on the Back button returns to the previous point of exploration and restores the previous filter settings and the last selection, so that you can easily continue reviewing the next e-mail message!
- If you are using a filter to focus on videos or documents, and then you double-click a video or a document to see the video stills exported for that video or the embedded pictures in that document, respectively, the filter is automatically deactivated, too.
- When you are viewing video stills only, in a gallery, and you use the Backspace key or "Find parent object" menu command to navigate to the video that this still belongs to (e.g. in order to play that video), then any active filters will be turned off so that the video can actually be listed. A simple click on the Back button returns to the previous overview of stills, enables the previous filters again, and restores the last selected item, so that you can easily continue with the next still!
- This works analogously when systematically looking at e-mail attachments, if occasionally for relevant attachments you would like to view the containing e-mail message (and e.g. print it or include it in a report) and then return to the list of attachments.

For more information about column-based filters, please see the description of the respective column.

3.3.5 Columns and Column-based Filters

Most filters and several columns are available with a forensic license only.

Name — Name of the listed file or directory and (only with a forensic license, only for directories and files with child objects) in parentheses in a different color optionall the total number of contained files in the volume snapshot. Allows to filter based on one or multiple filename masks, one per line. This filter is useful if you have a list of relevant filenames or keywords and want to find out quickly whether files with such names are present.

There are two different ways how to use the Name filter. The first way is to match certain expressions against the full name. The expressions may contain asterisks (wildcards), like **.jpg*. Up to two asterisks are allowed per mask if they are located at the beginning and the end of it. You may *exclude* files using file masks that start with a colon (:). Example: All files with names that start with the letter "A", but do not contain the word "garden": *A** in one line and *:*garden** in another. When multiple positive file mask expressions are used, they are combined with a logical OR, negative expressions (:) with a logical AND.

If the "Substring search in filename" option is active, then all the rules above do not apply. Instead, a search is run *within* the filenames for the specified characters or optionally GREP expressions. For example, just type "invoice" to find files whose filename contains the word invoice, not "*invoice*". For an explanation of GREP notation please see Search Options. The anchor $ does not work in this context.

The amount of text that can be pasted into the Name filter has been extended to 2 million characters in v17.7 (30,000 before). That doesn't mean that X-Ways Forensics can efficiently use a filter with many ten thousands of characters or more. When in doubt, use the "Match against full name" option, not the substring search, for better performance.

If an original name is found for a file in the Windows recycle bin or in an iPhone backup or certain other files during metadata extraction, that name is displayed in the Name column with the current unique name in square brackets. The current unique name is now also shown in square brackets in the case report. Both names are targeted by the Name filter.

The header of the Name column allows to quickly tag or untag all listed items with a single mouse click. It also indicates whether among the listed items are any tagged or untagged items.

Ext. — Filename extension. The part of the filename that follows the last dot, if any, except

if the last dot is the very first character (not uncommon in the Unix/Linux world).

Type — (Forensic license only.) File type. If the header signature of a file was not specifically checked (see Refine Volume Snaphot), this is merely a repetition of the filename extension and displayed in gray. Otherwise, if the file signature verification revealed the true nature of the file, a typical extension of that type will be output. That extension will be displayed in black if it is still the same as the actual extension of the file, or in blue if the actual extension does not match the type of the file. A convenient filter can be activated based on this column. In the filter dialog you can select individual file types or entire categories. You can load and save your selection. There are buttons that allow to expand or collapse all categories at once. Expanding all categories can be useful if you would like to quickly find a certain file type by typing its letters while the tree view window has the input focus.

Please note that collisions among file type designations become apparent when selections for the file type filter are loaded from .settings files or cases. For example if you had originally selected "mmf" = "MailMessage File" (category e-mail), then you will find that "mmf" is also selected as "Yamaha SMAF" (category Sound/Music). This is normal and does not change what the Type filter does. When in doubt, the Type filter also includes other types with the same designation, to avoid that anything is overlooked.

Type status — (Forensic license only.) The **status** of the Type column. Initially “not verified”. After verifying file types based on signatures (as part of refining the volume snapshot or viewing files in preview or gallery mode): If a file is very small (less than 8 bytes), the status is “irrelevant”. If neither the extension nor the signature of a given file is known to the file type signature database, the status is “not in list”. If the signature matches the extension according to the database, the status is “confirmed”. If the extension is referenced in the database, yet the signature actually found in the file is unknown, the status is “not confirmed”. If the signature matches a certain file type in the database, however the extension matches a different file type or there is no extension at all, the status is “newly identified”. Filter available.

Additionally, this column may contain a hint about the **consistency** of the format of files of various supported types as either “OK” or “corrupt”, for carved files perhaps immediately, for other files perhaps after file type verification or metadata extraction have taken place.

For an explanation of file type ranks and groups please see the description of File Type Categories.txt.

Type description — Displays the name of the application that a file type belongs to, what the filename extension stands for, etc. as specified in File Type Categories.txt. If the same extension occurs multiple times in the definition file, all its meanings are listed. For example, .pm could be a Perl module, a PageMaker document, or Pegasus file, or an X11 Pixmap file. (forensic license only)

Category	File type category corresponding to the file type, according to the definition in “File Type Categories.txt” (see below). Filter available. If the same file type/extension is defined multiple times, belonging to different categories, only one category for this file type will be displayed. The category filter works nonetheless. The category filter can be activated using a popup menu. In that popup menu you can also see statistics about the how many files of each category are currently listed in the directory browser (or would be listed if the category filter was turned off).
Evidence object	The name of the evidence object that the file or directory is part of. Useful in a recursive case root listing, i.e. when the directory browser shows all files of all evidence objects. (forensic license only)
Path	Path of the file or directory, starting with a backward slash, based on a volume's root. Filter available. The filter expressions are interpreted as substrings that can match any part of the path, so no wildcards are needed or supported.
Parent name, Child objects	Both columns come with filters. The filter for child object allows you for example to quickly find all e-mails that have an attachment with a certain name. The filter for parent name for example allows you to quickly find all attachments that were attached to e-mail with a subject that contains certain words. Note that filters for the columns Name, Parent name, and Child objects share the same settings and are mutually exclusive (cannot be active at the same time, one will deactivate the other). (forensic license only)
Size	Logical size of the file (i.e. size without slack) or physical size of a directory. Physical file size and valid data length (for files stored in an NTFS file system) can be seen in the Info Pane in File mode instead. If recursive selection statistics are enabled, with a forensic license the size of a directory is the total size of all the files directly or indirectly contained in that directory, otherwise the size of the data structures of the directory. Filter available.
Created	The date and time the file or directory was created on the volume it resides on. Not available on Linux filesystems.
Modified	The date and time the file or directory was last modified. On FAT, time precision is 2-second intervals only. On CDFS, the only available date and time stamp is listed in this column altough it does not necessarily indicate last modification. Filter available.
Accessed	The date and time the file or directory was last read or otherwise accessed. NTFS last access timestamps are displayed in gray if identical to the creation timestamp, as that on most systems likely means that these timestamps are simply not maintained, for performance reasons, and thus not very significant. On FAT, only the date is recorded. Filter available.
Record changed	The date and time the file's or directory's FILE record (on NTFS) or inode (Linux filesystems) was last modified. These are filesystem data structures that contain the file's meta data. Filter available.
Deleted	The date and time the file or directory was deleted. Available generally on Linux filesystems and possibly on NTFS (after a particular thorough file system data structure search and viewing/previewing the $UsnJrnl:$J file on the volume, if there is any). Not to be confused with so-called deletion timestamps that other forensic

tools may show you on NTFS volumes, for files that have not even been deleted from the file system. Filter available.

Content created — Creation timestamp that can be extracted from the internally stored metadata in various file types (see context menu command), as put there by the program that created the file. Internal timestamps are usually less volatile and can be more difficult to manipulate than file system level timestamps. They are useful for example for corroboration. Filter available. (forensic license only)

Timestamp columns designated with a superscript 2 (specialist license or higher) contain alternative timestamps. In the case of NTFS these values are taken from 0x30 attributes and represent previously valid timestamps from when a file was last renamed or moved, or possibly before some backdating operation occurred. Backdating operations are often applied by setup programs and also Windows itself (the infamous creation timestamp tunnelling effect, cf. http://support.microsoft.com/kb/172190), and of course potentially by ordinary application programs as well as by users for various legitimate or less noble purposes. Note that these columns are populated only if these previously valid timestamps are actually different from their current counterparts, and additionally Modified² and Record changed² only if different from Created², to avoid cluttering the screen unnecessarily with redundant information. That means any ² timestamps that you see there actually contain additional information and are not redundant.

Created² is also populated for HFS+ file systems, with the relatively new "Added date" timestamp from Mac OS X Lion and later as well as iOS, where available and if different from the regular Created date. That timestamp specifies when a file was added to the particular directory in which it is contained, even if originally created earlier.

The combined filter for all the timestamp columns allows to filter for certain date ranges (typical application) or for mere times, matching any possible date. For example if you are interested in unusual activity occurring in the middle of the night when the rightful office computer user is not working, you could filter for times such as between 22:00:00 and 05:59:59 (on a 24-hour clock). Obviously, selecting the right local time zone for the timestamp filter is crucial for this.

Please note that for FAT volumes, all timestamps are displayed as they are stored, in local time (they are not adjusted). For all other file systems the time zone concept applies.

Timestamps in the normal directory browser that meet the timestamp filter condition are highlighted. Timestamps in an event list that are identical to the event timestamp are also highlighted.

Attr. — DOS/Windows attributes on FAT/NTFS filesystems, Unix/Linux permissions and filemode on Unix/Linux/Mac filesystems, plus some proprietary symbols that are explained in the legend (forensic license only) and in topic 2.9.
“Partial initialization” means that according to the file system (NTFS or exFAT) the so-called valid data length is smaller than the logical file size, i.e. the data at the end of the file is undefined, similar to file slack has nothing to do with the file, and was

stored on the disk at that location before. You can see the valid data length of the file in File mode in the Info Pane, and the undefined area is highlighted in a different color.
When sorting by the Attr. column, files with “more interesting” attributes are listed first, e.g. attributes that indicate encryption, and files without any attributes set or whose attributes are unknown are listed last.
Filter available.

1st sector — The number of the sector that contains the beginning file the file's or directory's data. Sorting by 1st sectors means to sort by physical location on the disk and will show files next to each other, that are physically stored near to each other.. A filter is available, which allows to focus on files whose contents start in certain sector ranges, for example to identify files that are definitely affected by known bad sectors or to identify files whose contents are stored past the end of a known incomplete image. Remember that optionally you can see physical (disk-based) sector numbers here instead of logical (partition-based) sector numbers if so desired, see Directory Browser Options. The filter also allows to focus on carved files that are either aligned at sector boundaries or not, for example after having run a file header signature search at the byte level, to remove garbage files, which are more frequent among files that are not aligned.

ID — The identifier assigned to the file or directory by the file system or by WinHex. Not necessarily unique. A filter is available, which makes it more convenient to find other hard links of a given file.

Int. ID — The unique internal identifier of a file or directory in the volume snapshot. Items added to a volume snapshot last have the highest identifiers. Filter available. Useful for example and very easy to use if you would like to focus on the *x* files that were added to the volume snapshot last (after having refined it) or if you would like to resume a logical search with internal ID *y* (filtering out files that may have already been searched before).

For evidence objects that contain a huge number of files, the modulo option allows you to focus on a subset of files that is more or less representative of all files (though less random than files listed first when sorting by hash value). Applying the modulo operation to the internal ID will pick files from any directory, with any name, creation date etc. To see only 1,000 out of 100,000 files, i.e. every 100th file, use the operation "internal ID modulo 100 = 0". Also useful for testing purposes: If you wish to compare the performance of different hard disks, RAID systems, processors, configurations for volume snapshot refinements, you don't have to process all files in an evidence object. You can get quicker, yet likely representative results for example in 1/10 of the time if you only process every 10th file, pseudo-randomly selected by internal ID.

Even for normal work, examiners may not be required by their bosses/their prosecutor to conduct a 100% complete examination, for example if after review of a reasonably sized and representative subset you can extrapolate that about 10% of several 10,000 photos is illegal material.

Int. parent — The unique internal identifier of the parent directory of a file or directory in the

volume snapshot. Useful e.g. when exporting files and directories and there are multiple directories with the same name in the same path (e.g. one existing, one deleted), so that via the internal parent ID you can tell which file resided in which directory even if the path is ambigous.

Unique ID — An internal identifier of a file or directory that is unique within the entire case, not just within the volume snapshot of one evidence object, and unique for the whole life time of the case. When creating a new case, you have a choice between easily readable unique IDs that contain a delimiter (separating evidence object ID and int. ID) or a completely numeric ID (which may be better usable for some external programs when exporting a list of files). (forensic license only)

Owner — The ID of the owner of the file or directory, on file systems that record that information. On NTFS it's the SID, or, if X-Ways Forensics can resolve it to a username with the help of the SAM registry files already encountered while working with the case, the username. (forensic license only) Filter available.

Author — Shows the names of the authors of documents of various types (MS Office, OpenOffice/LibreOffice, RTF, PDF, ...), after metadata extraction. Filter available. (forensic license only)

Sender, Recipient — These columns are populated for e-mail messages and attachments extracted by X-Ways Forensics from e-mail archives, plus for original .eml files if metadata has been extracted from them. They come with filters. that allow you to enter any part of an e-mail address or name to search for certain e-mail messages. The filter expression is interpreted as a substring, so no wildcards are needed or supported. (forensic license only)

Link count — The hard link count of the file or directory, i.e. how often it is referenced by a directory. (forensic license only)

A hard link that just provides a short filename (SFN) to satisfy the legacy 8.3 requirements of old Microsoft DOS/Windows versions is not counted as a hard link. Instead, such files get their hard link count marked with a ° in the Links column of the directory browser. That way, the hard link count more accurately reflects the hard links actually present in the volume snapshot of X-Ways Forensics, and normal files always have a count of 1, whereas 2 or more means something more special. If a hard link count of 1 is marked with an asterisk (*), that means that the file or directory is stored as hard-linked in the directory structure in HFS+ although it would not be necessary based on the hard link count. If the hard link count is grayed out, that designates files that will be optionally omitted during a logical search to avoid unnecessary duplicate search efforts and duplicate search hits.

File count — The total number of files contained in a directory or in a file with child objects, in the volume snapshot, recursively, i.e. inclusive of further subdirectories. This number can also be found in the name column in parenthesis (depending on the settings). Computed only with a forensic license.

Term count — The number of search terms (not search hits) that have been found in a file. This takes into account all search terms ever used in simultaneous searches in a case, not for only the search terms that may have been selected in the search term list, unless

you have deleted search hits. You can sort by this column to get files listed first that are likely more relevant (because they contain more of the search terms that you were looking for). This column is populated only for evidence objects of a case. (forensic license only)

Search terms — Lists up to 25 of the search terms found in a file, those that are counted in the preceding column. Useful to get an idea of the search hits in a file even in the normal directory browser, without the need to switch to a search hit list. (forensic license only) Filter available, which is not limited to the 25 search terms displayed in this column.

Page count — The page count is extracted from PDF and some Office file types as part of metadata extraction and shown in this column. (forensic license only)

Pixels — The roughly rounded dimensions of a picture in thousand pixels (KP) or million pixels (MP, megapixels), as the result of width times height, for efficiency reasons stored as a very low precision value. The dimensions are computed simultaneously with skin color percentages, plus when viewing pictures (full-screen mode, preview mode, or in the gallery). Allows to easily distinguish between e.g. small browser cache garbage graphics and high-quality digital photos, with the associated filter, which allows you to focus on pictures with less or equal to the number of pixels that you specify or more or equal or both at the same time. (Works only approximately because of the low precision storage of pixel numbers.) Once at least 1 video still has been exported from a video file, the approximate resolution of the video can also be seen in this column. (forensic license only)

Analysis — Combined column that shows FuzZyDoc matches for textual documents as well as PhotoDNA matches and the computed amount of skin tones in raster images (or the fact that a picture is a black & white or gray-scale picture or too small to contain any relevant graphical content). Available after refining the volume snapshot if the underlying technology is available. Sorting or filtering by this column is the most efficient way to discover traces of e.g. child pornography or search for scanned documents (gray scale or black & white pictures). Sorting by the Analysis column in descending order lists files with FuzZyDoc matches first (those files with the most confident matches for any hash set near the top, with lower percentages following), followed by PhotoDNA matches (showing the category names in an internal PhotoDNA hash database), followed by pictures with no PhotoDNA matches in descending order of their skin tone percentage. After that, irrelevant pictures are listed (picture with very small dimensions), and then files that are not pictures, and near the bottom black & white and gray scale pictures. Text color coding in that column now makes it easier to distinguish between different kinds of categorizations.

Hash — The file's hash value, if computed. Filter available with a specialist and forensic license. Allows to filter for files that have a hash value, do not have a hash value, whose hash values start with certain hex values (if you specify only the beginning of a hash value) or have a certain value (if you specify a complete hash value). This filter can compare the hash values of files to up to 4 hash values that the user supplies as hex ASCII. Quicker alternative to creating a small hash set in the hash database if you just wish to quickly find a few files, e.g. duplicates of files with a

known hash value that you can just copy from the hash column in the directory browser. The easiest way to use this filter when looking for duplicates of a file, which does not even require copy & paste of hash values, is to right-click a hash value of a given file in the directory browser in hex ASCII notation (not Base32) and invoke the "Filter by" command in the context menu.

The Hash column displays pseudo-hash values in light gray color until real hash values have been computed. Pseudo-hash values are based on the file metadata, not on the file contents. They are available instantly even for very large files. They allow you to list files in a random order just like when you sort by real hash values, but without having to invest time to compute real hash values first. Useful for example for triage, if you have limited time and just wish to quickly look at some randomly selected files in a large evidence object first (e.g. pictures in a gallery) to determine how relevant an evidence object might be.

Looking at files in a *random* order might give you a more complete and accurate impression of what is stored in an evidence object, because the first *x*% of the files listed are more varied and more representative of the evidence object as a whole if they are in a truly random order. If you sort by name or path or size or timestamps on the other hand, many of the files you see will likely be somewhat similar (created by the same application or by the operating system, by the same user, for a similar purpose, created or copied or received around the same time, same file format, ...), so with some bad luck you will only see irrelevant files even if there is an equally large group of relevant files. Remember that if you don't sort in the directory browser at all, the view is skewed as well, because you will see the files in the order in which they are referenced by the volume snapshot, which is more or less the order in which they are referenced by the file system and thus not random.

Sorting by hash values can be combined with any filter, for example to see only pictures larger than 1 MB in a random order or only files of a certain user. Pseudo-hashes are not guaranteed to be unique or even remain the same when you close and re-open the evidence object.

Which hash value out of potentially two hash values stored in the volume snapshot is displayed in the Hash column can be changed in the Directory Browser Options dialog. Either the primary hash value or the secondary hash value or both at the same time (if the box is half checked). The Hash column filter is applied to the hash type(s) that is/are currently displayed. Which hash type(s) is/are displayed in the Hash column can be seen in the column header.

Hash set — The names of the hash sets in the internal hash database in which the file's hash value was found. Up to 64 matches are returned. Filter available. The Hash Set column shows known matches for both internal hash databases simultaneously. The filter can be used to filter for selected hash sets of one of the databases at a time. The database to choose hash sets from can be selected in the filter dialog. (forensic license only)

Hash — The category of the hash set that the file's hash value, if available, belongs to. Either

category	"irrelevant", "notable", or blank. Filter available. Note to users with two internal hash databases: The Hash Category column shows only one category. If you assign the hash value of a certain file in one hash database to one category and the hash value of the same file in the other hash database to the other category, you will be warned once during matching and given exact information about which hash value in which hash sets in which hash databases are conflicting. The categorization as "notable" will prevail when in doubt. (forensic license only)
Report table	The name(s) of the report table(s) that the file or directory has been assigned to. Filter available. If the parent file of a file has been assigned to one or more report tables by the user, then this is pointed out in the "Report table" column for the child object as well, in light gray color and with an arrow, except if the child object has report table associations itself. Reminds the user that the parent was reviewed and marked as relevant already, which can spare him or her the extra step of navigating to the parent again. (forensic license only)
Comment	The free text comment that may have been assigned to the file or directory by the examiner. Filter available. (forensic license only)
Metadata	Metadata that can be extracted from files of various types with the context menu. Filter available. (forensic license only)

Additional columns for search hit lists: Physical/absolute offset, logical/relative offset, description on the nature of the search hit (code page/Unicode, whether in decoded text, whether in file slack), search hit with context preview. If the logical relative offset is printed in gray, that means the search hit was found in the decoded text and the offset is not an offset in the file, but in the decoded text.

Additional columns for event lists: Timestamp, event type, event type category, description.

3.4 Mode Buttons

When examining a logical drive, partition, or image file with a file system supported by WinHex, there are several buttons that determine the display in the lower half of the window, below the directory browser. Forensic licenses only.

Disk/Partition/Volume/Container

Previously labeled “Sectors”, this default view shows the binary data in all sectors of the disk/ partition/volume/container represented by the active data window as hexadecimal code, as text, or both. Offsets and sector numbers are relative to the start of the respective disk/partition/volume/ container.

File

Looks similar to Disk/Partition/Volume/Container mode, but shows only the clusters allocated to the file or directory that is currently selected in the directory browser, in the order as used by the

file, defragmented if fragmented, decompressed if compressed, with offsets relative to the beginning of the file. When switching from File mode to Partition/Volume mode, X-Ways Forensics will automatically point you to the offset from the point of view of the partition/volume that is equivalent to the offset within the file where the cursor was positioned last, even if the file is fragmented, if there is an equivalent position (not if the file is a compressed or virtual attached file or an extracted e-mail message or an exported video still etc.).

Preview

Checks the type of the file currently selected in the directory browser and displays the file with the help of the separate viewer component, except if the viewer component is not active or if it's a picture (supported file types see Gallery below) and the viewer component should not be used for pictures. Even incomplete pictures (e.g. files incompletely recovered because of fragmention) can usually be displayed partially. If the viewer component is not active and the file is not a picture in one of the supported formats, a rudimentary ASCII text extract from the beginning of the file is displayed.

Details

Contains all the information on a single selected file from all the directory browser columns, including those that are not currently visible. Very useful for example if the path is very long and does not fit on the screen in the path column, maybe not even in the path tooltip display. Also allows to easily copy the filename or file path or selected other data to the clipboard.

The Details mode also shows NTFS file permissions (stored in access control lists, ACLs). Each element has typically the property "Grant" or "Deny" and an SID to which the permission applies. The SID is translated into a friendly name if possible. The permission itself is either R = Read Permission, C = Change Permission, Full Control or Special Access. For a Special Access right, all individual rights are listed. For each permission there can be two inheritance flags: container inherit (CI), object inherit (OI) or two propagation flags: inherit only (IO), no-propagate inherit (NP). Usually the final list element is the group membership property.

The Details mode also extracts some essential internal metadata from OLE2 compound files (e.g. pre-2007 MS Office documents), MS Office 2007 XML, OpenOffice XML, StarOffice XML, HTML, MS Access, MDI, PDF, RTF, WRI, AOL PFC, ASF, WMV, WMA, MOV, MP4, 3GP, M4V, M4A, JPEG, BMP, EXE/DLL, JIDX (Java applet cache), THM, TIFF, GIF, PNG, GZ, ZIP, PF, IE cookies, DMP memory dumps, hiberfil.sys, PNF, SHD & SPL printer spool, RecentFilecache.bcf, WIM Vista image files, PhotoShop PSD, INDD (Adobe InDesign), DocumentSummary alternate data streams, tracking.log, .mdb MS Access database, manifest.mbdx/mbdb iPhone backup, IconCache.db. For MS Office documents, you will often see many more timestamps (e.g. Last Printed), subject, author, organization, keywords, total edit time, and much more.

Gallery

Checks the file signature of all the files in the currently visible portion of the directory browser. If found to be a picture, a thumbnail is displayed, otherwise a brief summary (filename, size, signature). By scrolling in the directory browser, the gallery view scrolls as well. You may

switch the directory even while the thumbnails are still loading. By double-clicking a thumbnail, you get a full-size view of a picture, where you may zoom in and out using the keys + and -. Even incomplete pictures (e.g. file incompletely recovered because of fragmention) can usually be displayed partially. Supported picture file types: JPEG, PNG, GIF, TIFF, BMP, PSD, HDR, PSP, SGI, PCX, CUT, PNM/PBM/PGM/PPM, ICO. Optionally, the gallery can also show files of other types as thumbnails, using the viewer component. The gallery does not go together very well with search hit lists.

When a View window displays a picture, if limited to one such window, that window will be updated with the next picture when you hit the cursor keys in the gallery. Useful especially if the View window is centered on the second monitor if the gallery is on the first monitor, on a spanned desktop. Avoids having to press the Enter key to view the picture and another key to close the View window to get the input focus back to the gallery.

Calendar

Gives a convenient visual overview of the timestamps of all listed files/directories, from all 6 timestamp columns of the directory browser, in the form of a calendar, or when in event list mode a similar overview of all listed event timestamps. Each day with at least one time stamp is marked in the calendar with a gray color. The more activity on a day, the darker the color. Weekends (Saturdays and Sundays) are specially marked with x. Hover the mouse over a day to find out how many timestamps exactly fall into that day. Left-click a day to select that day as the left boundary of the timestamp filter, or right-click it to define it as a right boundary. Middle-click a day to filter for timestamps on that particular day only. If the same file is listed more than once (which can happen in a search hit list if it contains more than 1 search hit), then its timestamps are also represented more than once in the calendar.

When not showing events, you can now decide which column's timestamp should be included in the calendar. Columns that are hidden (have a width of 0 pixels) are excluded, all other columns are included. The status bar reminds you which columns are included even if not currently visible because of horizontal scrolling.

Years in the calendar with no timestamps are grayed out. The number of a year is displayed in a darker shade of gray the more timestamps are listed for that. All shades of gray try to give the examiner a better and quicker impression of peaks or absence of activity.

Example: During which period of time were most JPEG files processed on a volume? Right-click the root directory in the directory tree (case data window) to recursively list all files from all subdirectories, then use the file type filter to limit the view to JPEG files, enable the calendar view.

Raw

In Preview mode, in conjunction with the viewer component, Raw mode renders the file as plain text. This can be useful for example for HTML files to see the HTML source code, for .eml files to the see complete e-mail header, and generally when in search hit list mode the viewer component cannot highlight a search hit in Preview mode (because then it might contained in metadata or control code that would be represented in raw Preview mode, but not normal Preview

mode). You can make Raw preview mode persistent by holding the Shift key when activating Raw mode.

Sync

Synchronizes the directory browser and the directory tree in that when in a recursive view you select a file in the directory browser, its parent directory will be highlighted. Also when *clicking* the Sync button, unless the volume snapshot was created without cluster allocation information (see Security Options), the file that occupies the currently displayed sector in Volume/Partition mode will be automatically selected.

Exploration Mode

Button with a curly turquoise arrow. Toggles between normal and recursive exploration of a directory. When exploring recursively, you do not only see the contents of the current directory, but also the contents of all its subdirectories and their subdirectories, and so forth. To explore a directory recursively, you may also right-click it in the directory tree.

Multi-monitor support

It is possible to detach the lower half of a data window (with Disk/Partition/Volume mode, File mode, Preview, Gallery etc.) from the data window, by clicking the three dots that are located left to the mode buttons. After that, you can freely move and resize it on the screen. On multi-monitor this allows you to have that part of the user interface on a separate screen and even maximize it there. Reintegrating it into the main window is done by clicking the same three dots again or by clicking the Minimize button.

3.5 Status Bar

The status bar displays the following information about a file:

1. Number of current page and total number of pages (disk editor: sectors)
2. Current position (offset)
3. Decimal translation of the hex values at the current position
4. Beginning and end of the current block (if currently defined)
5. Size of current block in bytes (ditto)

Click the status bar cells in order to...
1. Move to another page/sector,
2. Move to another offset,
3. Define the integer type for decimal translation and
4. Define the block.

Right-click the status bar in order to copy pieces of information from the status bar into the clipboard.

Right-clicking the 2nd status bar field allows switching between absolute (default) and relative offset presentation. This is useful when examining data that consists of records of a fixed length. After specifying the record length in bytes, the status bar displays the current record number and the relative offset therein.

Right-clicking the 3rd status bar field allows copying the four hex values at the current position in reverse order into the clipboard. This is useful for following pointers.

3.6 Data Interpreter

The Data Interpreter is a small window that offers possible translations for the data at the current cursor position. Whether it is shown or not can be controlled via the View menu, not with the options of the data interpreter. Contrary to popular belief among some WinHex users, it totally disregards any block if selected and always interprets from the byte where the cursor is. The options dialog lets you specify the data types to interpret. These are various integer data types (by default in decimal notation, optionally hexadecimal or octal), the binary format (8, 16 or 32 bits of a byte), four floating-point data types, assembler opcodes (Intel®), and date types.

The Data Interpreter can interpret UNIX/C, Java/BlackBerry/Android and Mac Absolute timestamps stored as decimal ASCII text instead of in binary. You will find a context menu item for that as well as a checkbox in the options dialog. The Data Interpreter optionally translates timestamps of all formats except MS-DOS date & time to local time (the time zone defined in the General Options). You will find a context menu item for that as well as a checkbox in the option dialog.

The Data Interpreter is also capable of translating most data types back into hex values. Make sure a file is open in an edit mode other than read-only mode, enter a new value in the Data Interpreter, and press **ENTER**. The Data Interpreter will then enter the corresponding hex values into the edit window at the current cursor position.

Right-click the data interpreter to bring up a context menu. This will let you switch between big-endian and little-endian translation of integer and floating-point data. You may also choose between decimal, octal, or hexadecimal integer representation. See the Data Interpreter Options for more settings.

The decomposition of V1 GUIDs into timestamp, sequence number and MAC address in the Data Interpreter as well as in templates is optional. In the Data Interpreter options you can now choose to force the decomposition (fully checked) or prevent it (to always get the standard GUID notation with braces) or to see the decomposition only if the timestamp is not too implausible (half checked). The latter setting is helpful for example for Apple GPT values that claim to be V1 GUIDs, but contain twisted ASCII text instead of valid timestamps.

Hints:

- Some hex values cannot be translated into floating-point numbers. For these hex values the Data Interpreter displays NAN (**n**ot **a n**umber).

- Some hex values cannot be translated into valid dates. The value ranges of different date types are more or less narrow.
- There are redundancies in the Intel® instruction set, which show up in the Data Interpreter as duplication of both hex opcodes and mnemonics. Floating-point instructions are generally displayed as F***. More detailed reference can be found in the Intel® Architecture Software Developer's Manual Volume 2: Instruction Set Reference, available in PDF format on the Internet.

3.7 Position Manager

The Position Manager maintains a list of file or disk offsets and corresponding descriptions, also called *annotations*. It is also used for search hits when not working with a case, but *much* less powerful than a search hit list. Navigating from one entry to the next is easy if you press Ctrl+Left and Ctrl+Right. You may enter new positions and edit or delete existing entries. If a special offset in a file is important to you, you can add it to the Position Manager. This makes it a lot easier to find it again later, and you do not have to remember it. Descriptions may be up to 8192 characters in size. An appropriate description for instance could be "Data chunk begins here!". Optionally all positions maintained by the Position Manager can be *highlighted* in the editor window in a unique color you specify, and their descriptions displayed in yellow tooltip windows when the mouse cursor is moved over them. You may also add or edit positions with the context menu of an edit window or by clicking the middle mouse button in an edit window.

Click the right mouse button in order to see a context menu in the Position Manager. The context menu provides additional commands. You may delete, load or save positions, even export the list as HTML. If the position list in the *general* Position Manager was changed, it is saved in the file *WinHex.pos* when exiting WinHex, so that they are still available in the next session. Only search hits are not permanently saved, unless they have been edited via the context menu.

The complete documentation of the POS file format is available from the WinHex homepage at http://www.x-ways.net/winhex/.

3.8 Useful Hints

- Menu commands that affect individual, selected items in the directory browser or in a search hit or bookmark list can be found in the context menu that opens when you right-click such items. You won't find such commands in the main menu.
- Use the mouse buttons as follows to define the block (if the context menu is switched off):
  - Double-clicking left sets the block beginning.
  - Single-clicking right sets the block end.
  - Double-clicking the right button clears the block.
- You may want to define the block using the keyboard (**SHIFT**+arrow keys or **ALT**+**1** and **ALT**+**2**).
- Use the **TAB** key to switch between hexadecimal and text mode.
- Use the **INS** key to switch between insert and overwrite mode.

- **ENTER** displays the Start Center.
- **ESC** aborts the current operation if any, otherwise clears the block, dismisses an active dialog or template window.
- **PAUSE** stops or continues the current operation.
- **F11** repeats the last Go To Offset command. **CTRL**+**F11** works in the opposite direction (from the current position).
- **ALT**++ is a variant of the Go To Offset command specifically to jump a certain number of sectors *down*.
- **ALT**+- is another variant specifically to jump a certain number of sectors *up*.
- **SHIFT**+**F7** switches between three character sets.
- (**SHIFT**+)**ALT**+**F11** repeats the last Move Block command.
- **CTRL**+**SHIFT**+**M** invokes an open evidence object's annotations.
- **ALT**+**F2** recalculates the auto-hash (checksum or digest) after a file was modified.
- **ALT**+**LEFT** and **ALT**+**RIGHT** allow for switching between records within a template (just as the "<" and ">" buttons). **ALT**+**HOME** and **ALT**+**END** access the first and the last record, respectively.
- **ALT**+**G** moves the cursor in the edit window to the current template position and closes the template window.
- **CTRL**+**F9** opens the Access button menu (disk edit windows only).
  - Ability to specify how cooperative X-Ways Forensics behaves during long operations (e.g. hashing, searching) when competing with other processes for CPU time, by pressing Shift+ Ctrl+F5. 0 is the default setting (not specially cooperative). You could try values like 10, 25, 50, or 100 (maximum willingness to share CPU time) e.g. if X-Ways Forensics is executed simultaneously by different users on the same server, for a fairer distribution of CPU time.
- WinHex accepts filenames specified in the command line and is drag-and-drop capable.
- Use scripts to make your work with WinHex more efficient.
- You can specify the name of a script as a command line parameter.
- “Invalid input”: When clicking OK in a dialog box and getting the “Invalid input” error, pay attention to what control item in the dialog box is blinking, as the value in that item is the one that is not accepted.
- Switch from hexadecimal to decimal offset presentation by clicking the offset numbers.
- Try clicking the status bar cells (left and right mouse button).

Since the days of Windows 95 (or perhaps even Windows 3.1?) users can press Ctrl+C to produce a plain-text representation of standard Windows message boxes in the clipboard. With message boxes in WinHex and X-Ways Forensics it works the same. Although this is an elementary feature in Windows for more than 20 years already and should be known to any experienced Windows user and although WinHex and X-Ways Forensics make users aware of that ("Did you know? ..."), the great majority of users for some reason still take graphical screenshots of message boxes and paste them into HTML e-mails, for example when they report error messages, although that is more work than simply pressing Ctrl+C and Ctrl+V and although it inflates the size of the e-mail unnecessarily, as a few ASCII characters need much less space them thousands of pixel values. That also means the screenshot will get lost if the e-mail is converted to plain text when being replied on, and of course the error message text will not be searchable in a graphical screenshot and cannot be conveniently selected and copied to the

clipboard as text by the recipient, and the recipient cannot be sure of the exact Unicode value of certain characters for which multiple variants exist.

In WinHex and X-Ways Forensics it is even possible to copy a rudimentary ASCII representation of dialog boxes and almost all their control items (static text, push buttons, check boxes, radio buttons, list boxes, combo boxes, and tree view controls) including their states (unchecked, checked, half checked) by pressing Ctrl+C with an active dialog box on the screen (not if an edit box with a selection has the input focus). There is also a dedicated command in the window menu of an dialog box. That menu is a.k.a. the system menu or control menu, and it pops up when right-clicking the title of a dialog box. This copy command is a very efficient way to show your settings in a certain dialog box to other users and let them copy strings for use in their own edit boxes, so that they don't have to type them, avoiding typos. The text representation is even more powerful than a screenshot because it shows the contents of edit boxes and list boxes completely, even if these controls have scrollbars and the contents exceed the physical boundaries of the controls on the screen. Unicode characters are supported. We suggest that users take screenshots of message boxes and dialog boxes only if absolutely necessary, for example if they wish to graphically highlight certain control items in a Photoshop or similar programs to get the message across.

Settings in practically all dialog boxes can also be conveniently saved to and loaded from files as needed, for example to share them with other users or for future use, via the system menu. This function can remember the selection states of the most important control types: check boxes, radio buttons, list boxes, combo boxes, and tree view controls. This works even if the controls are currently invisible. The settings are stored in files with the .dlg extension (for "dialog"), in the same directory as templates and scripts. The contents of edit boxes are also remembered. However, this function does not remember the contents/text labels of check boxes, list boxes, combo boxes, and tree view controls, e.g. which code page a check box represents in the Simultaneous Search dialog, which report tables exist in the Report Table filter list box, which external programs are listed in the Viewer Programs dialog window, which file types are listed in a tree view control etc. It also does not remember the order of controls or list items. It also does not remember settings in a dependent dialog window (which opens e.g. when clicking a "..." button). The functionality is not available for the Directory Browser Options dialog window. For the directory browser options please save and load .settings files by clicking the icons in the directory browser caption line. The functionality to store dialog window selections in files is very useful for example for the Export List command, where some users repeatedly need different settings for different purposes, and where the items in the list box are always the same (just the available columns), except after changing the language of the user interface.

4 Menu Reference

Note: Commands in the main menu (File, Edit, Search, ...) always apply to the active data window as a whole (which e.g. represents an open file or an open disk), or to files/disks that are still to be specified by the user. They never apply to the file(s) currently selected in the directory browser. That’s what the directory browser context menu is there for.

4.1 Directory Browser Context Menu

The directory browser context menu allows the user to directly interact with the currently *selected* files/directories, notably *not* the *tagged* items. There are a number of menu commands which are available depending on the selected items. Double-clicking files and directories will, depending on the circumstances, either invoke “View”, “Explore” or the associated external program.

View

This command allows viewing the selected file with WinHex' internal viewers for Windows Registry files and various graphical file formats. If the separate viewer component that comes with X-Ways Forensics is active, all other files are sent to that viewer. If it is not, the first installed external program will be called instead. Exceptions to all of the above are files beyond 2 GB in size and NTFS system files. These are always opened as data windows.

When viewing a file in a separate window, you may press (Ctrl+) Page Dn/Up to close the window and view the next file in the directory browser in a new window. If a View window displays a picture and viewing pictures if limited to one picture at a time, that window will be updated when you press the cursor keys in the gallery. Useful especially on a spanned desktop, if the View window is centered on the second monitor and if the gallery is on the first monitor. Avoids having to press the Enter key to view the picture and another key to close the View window to get the input focus back to the gallery.

Explore

Only available for directories and archives (ZIP, RAR, TAR...), this command allows navigating into them within the directory browser. Double-clicking archives or directories does the same. A command that allows listing the contents of directories as well as their subdirectories at the same time can be found in the directory tree's context menu instead (in the Case Data window, "Explore recursively").

Viewer Programs

Allows to send the selected file(s) to one of the external programs currently configured or the file's associated program in the current Windows installation. This association is determined based on file extension as is usual within Windows.
You also have the option to open files in an external program that you select ad hoc. The program that you select will be saved as a standard custom viewer program if you have not used all slots for external viewer programs yet, and then also remembered for next time when you invoke the same menu command.

Open

Opens currently selected files or directories in separate data windows. Unlike File | Open, where files can be opened just like in any other application with the help of the operating system, this is a forensically sound operation in that it does not update any timestamps etc. because the

operating system is circumvented and the logic to read the file's contents from the correct disk sectors is implemented in WinHex itself for various file systems. No changes can be made to files that were opened in this fashion, however. In the case of a directory, the directory's data structures will be opened.

Print

If the separate viewer component is active, you may select files for printing. Allows to print multiple selected documents without interruption/the need to click somewhere after each document, optionally along with child objects (e.g. e-mail attachments together with their respective e-mail message).. The optional cover page contains the date and time when the print job was started and selected meta-information, e.g. filename, path, evidence object title, file size, description, time stamps, comments, ... The cover page is printed by X-Ways Forensics itself, the following pages with the actual document are printed by the viewer component. Another option is to have X-Ways Forensics print the filename and path on the first page. This option is not bound by the same path length limitations as the header optionally printed by the viewer component. To avoid that the path is printed twice on the first page, have either X-Ways Forensics or the viewer component print it, not both. You can print *just* the cover page by choosing to print only the pages 0 through 0 of the document or picture itself. The header line of the cover page, which specifies which user and which program and version created the print job, is optional. Useful if you wish to show the printout to witnesses or the suspect who should not know the username of the examiner.

Recover/Copy

Allows to copy the selected files from their current location to a location available for a standard Windows file dialog, e.g. out of an interpreted image file or from a local disk. This can be applied to both existing and deleted files and directories. Illegal filename characters are filtered out.

If necessary, you can manually enter the output path by clicking the "..." button in the same line where the path is displayed. Useful if you wish to specify a network location that Windows does not list automatically.

Numerous extra features are available with a forensic license:

- The complete original path can optionally be recreated in the output directory, or optionally (if half checked) only a partial path. The evidence object name becomes part of the recreated path, too, if you either copy from within the case root or if you do not have X-Ways Forensics default to the evidence object folder as the output directory (see case properties). A partial path is the path starting from the currently explored directory, or when copying from the recursively explored case root window only the evidence object name, not the path within the evidence object.
- Overlong paths are supported (more than 260, up to 510 characters, for output path + optional original path + original filename). You can still limit paths to the ordinary length of 260 characters or less if you would not be able to access (e.g. view, copy or delete) such files otherwise (because ordinary tools like the Windows Explorer do not allow that). If the output path of a selected file exceeds the limit, the name of the is shortened until it fits. If shortening the name does not help to stay under the specified path length

limit, the file not copied, but added to a report table, so that you can conveniently select all the omitted files later and copy them seperately without original path if you like.

- An option exists to name output files after their unique ID, while preserving the filename extension. If only half checked, the files will not be named purely after the unique ID (+extension), instead, the unique ID will be *inserted*, between base filename and filename extension.
- Files that could not be copied (e.g. if path too long) are added to a report table.
- The original timestamps (creation, modification, last access, if available) are re-applied to the recovered/copied files.
- Duplicate filenames will be changed to unique filenames by inserting incrementing numbers before the extension. So if you copy all files to the same directory, even those from different evidence object, all output filenames will be unique (and the copylog file allows you to later find out which file was originally named how and originated from where and which metadata it had).
- The presumed correct file type of newly identified files, if different from the extension in the original filename or if the filename does not have any extension, can optionally be appended to the output filename. This option also has an effect when copying files to view them with the associated program.
- When working with an active case and if special logging for this command is enabled, the copy/recovery process is documented in the file “copylog.html” or “copylog.txt”. All available metadata and the output filename (optionally including target path) can be recorded. The file can be created either in the _log subdirectory of the case or in the Recover/Copy target folder. Cf. also Case Properties.
- Slack space can optionally be included in the output, either as part of the file or separately, or *solely* slack can be copied.
- You can choose whether to also copy child objects of selected files, of any kind of child objects (if fully checked) or only e-mail attachments (if half checked).
- You can also choose whether to copy files that are filtered out.
- If you have X-Ways Forensics recreate the original path for copied files, the hierarchical location of files that are child objects of other files must be reflected appropriately, too. And that must happen with the help of a directory, because ordinary file systems do not support the concept that a file can contain further files, as is normal with volume snapshots in X-Ways Forensics. However, there would be a name conflict if an artificial directory was created with the same name as the parent file, as that parent file might be selected for copying as well, and would of course be created in the same directory as the aforementioned artificial directory that is needed to reflect the path of the child object. Hence the artificial directory must be named slightly differently. It can be truncated after a user-defined number of characters, and this is useful in particular for e-mail messages that are named after the subject line and of course can contain attachments as child objects, to avoid overlong paths. Also either a single suffix character of your choice is appended (and by default that is a special Unicode character that is invisible in complete Unicode fonts, such that the directory seems to have exactly the same name as the corresponding parent file), or otherwise some descriptive words like " child objects" are appended to the name (but that unfortunately increases the total path length, which all too often exceeds common limits). If the edit box for the suffix character seems to be blank, that is most likely because the aforementioned invisible Unicode character is in there. It has a width of 0. To replace it with any other character, remove the invisible character

first, by clicking in the edit box and hitting the backspace key on your keyboard.

- Existing and deleted objects can be grouped together in separate output directories named “Ex” and “Del”.
- Further grouping/classification of copied files in separate directories based on selected directory browser columns is supported: description, file type, file type description, file type category, sender, owner, hash set, hash category, report table associations, search terms.
- If both an attachment and the corresponding e-mail message (its parent) are selected for copying and not excluded by filters, the attachment can optionally be embedded in the resulting output .eml file as Base64 code instead of copied separately. That facilitates viewing the complete e-mail including attachments. To view .eml files you can use Outlook Express, Windows Mail, Windows Live Mail or Thunderbird (all free of charge). If certain attachments cannot be embedded, you will be informed via the Messages window, and in such a case they will be copied separately, as if the embedding option was not selected.
- NTFS alternative data streams (ADS) can optionally be output as ADS. By default, they are recreated as ordinary files, to make them more easily accessible.
- You may use the alternative names of files, if available, for the output. The alternative name, if one exists, can be seen in the directory browser in square brackets. For example, when parsing iPhone backups, X-Ways Forensics automatically changes artificial generic filenames back to what they were originally. Or, when parsing $I files from the Windows recycle bin, the corresponding $R files are given their original names. If for some reason you prefer the untranslated filenames when copying such files off the image to your own hard disk, for example because you wish to process these files with some external tool that expects the artificial filenames, then you can now use this option.

When using the Recover/Copy command in search hit lists, directories that contain hits are recreated in the output folder as files, as the user likely wishes to retain the original data that contain the actual search hit. Child objects are never copied along with their parent objects from within a search hit list.

Export List

Requires a specialist license. Exports data about the selected items in the directory browser to a tab-delimited text file or to an HTML file, which can be easily viewed in any web browser, also imported and further processed e.g. in MS Excel and MS Word. A third option (except for search hit lists) is an XML file. The list can alternatively be copied into the clipboard in the format as chosen, for example to paste it directly into an externally edited report. The columns to export are freely selectable. Even the search hit column can be exported, with the textual context around each and every actual hit, where the search term itself can be visually highlighted with a yellow background color (not recommended for output to MS Excel). You may choose to split up the result into multiple files for example to avoid a huge HTML file that Internet browsers will choke on.

There is an option to copy files off the disk/image and link them from the HTML output. The links can be found in the Name column. The behavior is affected by two case report options: "Name output files after unique ID" and "Embed attachments in parent .eml file". This option presents an interesting layout alternative to the regular output of report tables and also an

alternative to the Recover/Copy command.

Report Table Association

for Report Tables, see above

Edit Comment

Requires a forensic license. Use this command to add a comment to an item in the directory browser or to edit or remove an existing comment. After entering comments, you can conveniently set the filter such that only commented items are shown or only items with specific comments, e.g. those with a certain relevance.

Edit Metadata

Requires a forensic license. Allows to edit the metadata field of a file once metadata was extracted. Useful if you wish to include selected metadata (not all extracted metadata) in a report.

Refine Volume Snapshot and **Simultaneous Search** in items that are *selected* in the directory browser

Tag/Untag Item

Requires a forensic license. Tagging files means highlighting them visually (placing a blue square at the beginning of a directory browser item), for various reasons, e.g. to mark them as relevant, or memorize a position in a sorted list, or to limit volume snapshot refinements to tagged files. *Tagging* is not to be confused with *selecting*.

Exclude/Include

You may exclude selected items or all tagged or all untagged items. If actually filtered out, excluded files are omitted from the directory browser, the gallery view, and all commands that can be run from the directory browser context menu. If you are only allowed to examine the contents of certain directories, you could initially exclude all files in all other directories to ensure that. Refining the volume snapshot can be limited to files that are not excluded. Excluded items are actually filtered out only if the corresponding filter is enabled in the directory browser options. If not filtered out, they are listed in gray and can be included again with the directory browser context menu.

If you wish to review files with identical contents only once and if filenames, timestamps, deletion status and other file system level metadata are of secondary relevance, then you can use the command Exclude | "**Listed duplicates based on hash**" to identify duplicates among all the currently listed files (it says listed, not selected!), based on hash values (if hash values were calculated!) and exclude them. Only one file in each group of identical files will not be excluded. Do not apply this command more than once to the same files, or else *all* identical files might be excluded, depending on the sort criteria.
Special rules: When in doubt, this function chooses to keep existing (not deleted) files, and

among deleted files rather discards carved files and keeps files found via file system data structures.
Optional special rules: Identical e-mail messages with different attachments (child objects) will be marked as duplicates, but not excluded. Identical attachments (child objects) will be marked as duplicates, but they will be excluded only indirectly if they are part of identical e-mail messages and those are excluded, too. This facilitates the examination and also avoids a situation where the parent (e-mail message) of one e-mail+attachment family and the child object (attachment) of another family is excluded.
If later you find a relevant file for which there were duplicates and you are interested in the duplicates, too (e.g. in their filenames, paths, or timestamps), you could create a hash set of that files to conveniently and automatically identify all the duplicates, by matching the hash values of all files against that particular hash set and using the hash set filter.

You may also exclude files based on identical names instead of identical hash values. This is a case-insensitive comparison and of course should be used only if you know what you are doing, as it does not compare the file contents at all. Could be useful for example if you wish to get rid of multiple copies of the same files found in backups if you do not need to keep different versions of these files. If prior to the comparison for example you sort by last modification date in descending order, this will ensure that the newest version of the file will be kept and all older versions will be excluded. Files with identical names are not marked as duplicates in the Attr. column. That happens only if you identify identical files based on hash values.

In search hit lists you may
1) permanently delete *selected* search hits,
2) permanently delete *duplicate* search hits. Search hits are considered duplicates if they either have identical physical offsets or, if they don't have physical offsets, if their logical offsets and the corresponding internal file IDs are the same. When in doubt, X-Ways Forensics will keep the longer search hit (as "Smithsonian" for example is more specific than "Smith") and favors search hits in existing files.

Navigation

The Navigation group of commands allows interactions with the currently selected file on a generally more technical level. It allows to directly locate the data structure in the file system that defines a file (e.g. FILE record in NTFS, inode in Ext2/Ext3/Ext4, directory entry in FAT) and also to sort files by the offset of their defining data structures.

The Navigation menu also allows to produce a list of all the clusters allocated to the selected file or directory. From the context menu of that list window, the cluster list can be exported to a text file. Optionally the list can be shortened and its creation greatly accelerated by omitting clusters in the middle of a fragment. Omissions are indicated by ellipses. This option takes effect only when you produce a cluster list the next time.

Find parent object: Navigates to and selects the parent object of the selected object. Equivalent to pressing the Backspace key. The child object can be an ordinary file in a directory, or an e-mail message in an e-mail archive or a file attachment in an e-mail message or a picture in a document or a file in a compressed archive etc.

Find related object: This command allows you to conveniently navigate to the so-called related object if one exists for the selected file or directory. Alternatively, you can press Shift+Backspace.

See selected item in its directory: Will show you the selected file or directory among its siblings. Useful to quickly check out whether there are more notable files in the same directory or to better understand the function of the file when you see it in context.
See selected item from volume root: Will show you the selected file among all other files in the same volume, recursively explored from the root of that file system. Useful for example to see whether there are any files with the same name, the same ID (e.g. previous version from a volume shadow copy), same owner, same sender, or similar timestamps etc. in the same file system (just sort accordingly).
Both commands can be also be used from within the case root window and from within search hit lists (so the previous "Go to file in directory browser" command becomes obsolete). Remember you can click the Back button in the toolbar to conveniently return to the previous view.

Refine Volume Snapshot, Simultaneous Search, Run X-Tensions

These commands are known from the main menu. From the directory browser context menu they can be applied to the *selected* files.

Create Hash Set

Creates a hash set of the currently selected files and directories and their subdirectories directly within the internal hash database, either with ordinary file hash values or with block hash values or PhotoDNA hash values. For ordinary hash values there is an option to create multiple hash sets in a single step, where the hash values of the selected files are put into hash sets that are named after each file's report table association(s). This is useful if you categorize notable files in one case using report tables (e.g. based on different types of CP), and wish to quickly identify the same files again in other cases later, and automatically see the category that you had originally assigned, as the hash set name. The checkbox for that is labelled "Name after report table associations, if any". If a selected file does not have any report table association, its hash value will be assigned to the hash set named as you specify, just like if you do not check the new checkbox.

Attach External File/Dir

Requires a forensic license. Ability to attach one or more external files or a directory including subdirectories to the volume snapshot and have them processed by X-Ways Forensics like regular files in the volume snapshot. Useful if you need to translate, convert, or decrypt original files and would like to reintegrate the result back in the original volume snapshot, in the original path, for further examination, reporting, filtering, searches etc. Such external files will be completely managed by X-Ways Forensics once attached, copied to the internal evidence object subdirectory of the case, and marked as virtual files.

When attaching a single external file and holding the Shift key, X-Ways Forensics proposes a new name for that file that is based on the name of the file that is selected, and the attached file will be added to the same directory. Otherwise the external filenames of the files will be used and

they will become child objects of the selected object. It is still possible to rename virtual files in the volume snapshot later at any time.

When attaching an external directory to the volume snapshot, you are prompted whether the selected directory itself should also be attached or just its contents. Usually X-Ways Forensics creates virtual files in subdirectories in new virtual directories in the volume snapshot. There is, however, an option to accommodate the files in existing directories in the volume snapshot of the same name at the same position in the directory tree. Useful if you copy an entire directory structure off the image to convert/decrypt/translate/... files outside of X-Ways Forensics, and then want to bring the results back into the volume snapshot and see the edited files next to their original counterparts in the corresponding subdirectories. This can help for example if you wish to OCR and convert PDF documents that X-Ways Forensics has deemed non-searchable, using Adobe Acrobat.

Rename

Allows you to rename virtual directories and virtual attached files in a volume snapshot, or if the Shift key is pressed even ordinary files. Although the latter is not exactly forensically sound when dealing with original evidence, this can prove helpful in special situations, for example if a filename or directory name is too long to copy a file out of an image etc. The original filename will be kept as the alternative filename. Note that this does not rename the file in the file system (nothing is altered on the disk or in the image!), only in the volume snapshot, i.e. the internal database in X-Ways Forensics *about* the file system.

Specify type

Ability to specify the type of selected files yourself. Useful if you wish to identify types or subtypes in an individual way unknown to X-Ways Forensics, for example to be able to filter by these types later. For instance, how about categorizing TIFF pictures that are digitally stored faxes as type "fax"? Remember you can define your own file types in File Type Categories.txt.

Wipe securely

Files and directories that are selected in the directory browser can be securely wiped in WinHex (not X-Ways Forensics). The data in the logical portion of a file (i.e. excluding the file slack) and in clusters of a directory (e.g. containing INDX buffers in NTFS and directory entries in FAT) will be erased/overwritten with a hex value pattern of your choice. The existence status of the file in its file system will not be changed, i.e. it will not be marked as deleted, the clusters will not be released etc. No file system level metadata such as timestamps or attributes will updated because no operating system file level write commands are used. No file system data structures are changed, and no filenames will be erased, only the contents of files will be overwritten. Files that are compressed in archives or generally files within other files (e.g. e-mails and attachments in e-mail archives) cannot be erased. Previously existing files whose clusters are known to have been reused will not be erased. Note that by erasing deleted files you might erase data in clusters that belong to other files, so only select existing files if you want to avoid that (assuming consistent file systems). Also note that by erasing carved files you may erase too much or not enough data, depending on the detected file size and depending on whether the file was originally fragmented. And please note that wiping directories, i.e. erasing the data in the clusters allocated to a

directory, will cause existing files in that directory to become orphaned. More typically users only wipe the contents of files with this function, not the contents (data) of directories, if they still wish to use the file system.

Useful for example if copies of images are forwarded to investigators/examiners/other parties involved in a case who are not allowed to see the contents of certain files. Useful also if you have to return computer media on which child pornography has been found to the owner after clearing these files. Also useful if you are preparing images for training purposes that you would like to publish and if you would like to retroactively erase the contents of copyrighted files (e.g. operating system or application program files).

Both successfully erased files and files that could not be successfully erased will be added to separate report tables (when working with a case, with a forensic license only) by which you can filter to verify the result.

Filter for duplicates

Ability to filter for duplicates of a single selected file that are also currently listed in the directory browser, only if a hash value is available for the selected file and the other files. Actually filters for that hash value at that time, and thus does not depend on previous mass identification of duplicate files using the above-mentioned command Exclude | Duplicates in directory browser based on hash. In X-Ways Investigator the actual hash values are not displayed and cannot be computed, but they are imported from evidence file containers that come with hash values for files.

Mark hit as notable

In a search hit list, marks selected hits with a yellow flag and includes in them in the list of notable search hits. You may also press the space bar to mark a hit as notable or remove that mark. Holding the Shift key when invoking the menu command removes the "notable" flag from all selected search hits.

4.2 Data Window Context Menu

When you right-click the hex editor display (consisting of offset column, hex column, text column) of a file or a disk, you will get a context menu that allows you to define the boundaries of the block (start and end) and invoke a few more commands that apply to that block:

Add to User Search Hits: Forensic license only. Allows you to define search hits manually. Whenever you come across some relevant text, for example floating around in free space in Disk/Partition/Volume mode or within a certain file in File mode, you can select it as a block and right-click the block to add it as a so-called user search hit (i.e. some kind of search hit not found by the program). You can assign the search hit to an arbitrarily named search term/category. For example, if what you have found is related to suspect A, assign it as a search hit to a search term named after suspect A. If also related to suspect B, you can also assign it to another search term. You could also assign it to a real search term that you have used for an automatic search.

User search hits can be conveniently listed in and nicely exported from search hit lists just like ordinary (automatically generated) search hits. To distinguish then from ordinary search hits, in the search hit description column user search hits are marked with an asterisk (*). You can specify the correct code page for user search hits yourself when you define them, which may be essential to get the text displayed correctly. User search hits are stored related to an object in the volume snapshot if you define them in File mode. User search hits are forward compatible, i.e. older versions (v16.2 and later) can also see user search hits created by v16.6.

Add Block as Virtual File: Forensic license only. See Edit menu.

Add Position: Allows you to remember the position indicated by the currently defined block, either in the General Position Manager or in the Position Manager of the evidence object (when working with a case, if you right-click a block that is defined in an evidence object, forensic license only). Makes it easier to find the same position again later, and can be used to nicely highlight and explain (with tooltips) the structure of files or records of a certain format that you are analyzing/trying to reverse-engineer etc.

If search hits are highlighted in File mode (see General Options), you can also delete them via the context menu.

You can also get the complete Edit menu from here.

4.3 File Menu

New: This command is used to create a file. The file is principally opened in default edit mode. You have to specify the desired file size.

Open: Lets you open one or more files. You may choose an edit mode in case it is not predetermined in the Options menu.

Also allows to open physical disks, partitions and volumes as a file, by clicking a button labeled "Device..." in the file selection dialog. You can enter a device path such as
\\.\PhysicalDrive1 (for hard disk 1)
\\?\Volume{12345678-9abc-11a1-abcd-0123456789ab} (for a volume with that GUID)
\\.\C: (for a volume mounted as drive letter C:)
This functionality allows to open volumes that are not mounted as drive letters. To get an overview of volumes known to Windows, type "mountvol" in a command prompt window. You can also try to open exotic devices supported by Windows such as tapes and changers (not tested). Also this is how you can open alternate data streams whose path and name you know, which cannot be opened through the ordinary File | Open dialog, without opening the volume on which they reside.

Opening a hard disk as a file can be useful for example if you wish to clone that disk and if source and destination disk have different sector sizes (whether it makes sense in the first place to clone a hard disk despite the sector mismatch depends on the data). When treated as a file, there is no defined sector size and hence no possibility for a sector size mismatch. Device files can also

be interpreted as disks like images can.

Save: Saves the currently displayed file to the disk. In in-place edit mode, using this command is not necessary. When using the disk editor, this command is named “Save Sectors”.

Save As: Saves the currently displayed file under a different name.

Create Disk Image/Make Backup Copy: cf. “Images and Backups”

Create/Verify Skeleton Image: cf. “Skeleton Images”

Restore Image: Select an image that you would like to restore, i.e. whose sectors you would like to copy back to the original medium or some other medium, or select a or WinHex backup (.whx) file whose contents you would like to restore (could be a file or disk sectors). In the case of an image, the image will be preset as the source in the Clone Disk window (with a specialist license or higher, interpreted). Without a speciaIist license or higher, only WinHex backups can be restored if they are split.

Backup Manager: cf. “Backups”

Execute: Executes the current file if executable, or otherwise the associated program.

Print: Use this command to print a file, disk sectors or RAM contents. Define the printing range via offsets. You may select and set up a printer. Choose the character set for printing and accept or change the suggested font size. The recommended font size is calculated as follows: print resolution (e.g. 720 dpi) / 6 (e.g. = 120). If desired you may enter a comment which will be printed at the end.
In case you need more flexibility with printing, you can define a block and copy it using “Edit->Copy->Editor Display” as a hex-editor-formatted text into the clipboard. You may paste it in your favorite word processor. It should look perfect in “Courier New”, 10 pt.

Properties: Allows you edit the size, the time stamp and attributes of a file or a directory in your own Windows system, in WinHex only. Changeable attributes are: A (to be archived), S (system), H (hidden), R (read-only), X (not to be indexed), T (temporary), ~ (sparse). After entering new values in any area (size, timestamps or attributes), simply press the **ENTER** button to apply them. Click the button with the ellipsis to select a new file, or enter path and name directly into the edit box next to that button and press the **ENTER** key. The latter will also work for a directory.
Please note that setting or removing the sparse attribute does not necessarily change the allocation status of already assigned clusters, but will definitely have an effect on newly assigned clusters when you expand the file by setting a larger file size in the same dialog window.

Open Directory: Opens a window that represents a directory on your own computer and allows you to see all its files and subdirectories.

Open Files: This command is used open several files that meet special requirements at a time. Select a folder in which to open files. Subfolders are browsed optionally. You may specify a

series of file masks (like “w*.exe;x*.dll”). There is also a switch that permits opening only those files that contain a certain text or certain hex values. The standard search dialogs are displayed upon request for this purpose. If WinHex is not set up to work as a viewer or in-place editor (this can be done in the Tools menu), you may choose an edit mode.

Save Modified Files: All files which have been changed are written to the disk.

Save All Files: All files that have not been opened in view mode are written to the disk.

Exit: Use this command to end WinHex. You will be prompted to save any modifications to files and disks.

4.4 Edit Menu

Undo: Reverses the last modification, in case the corresponding undo option was activated.

Cut: Removes the current block from the file and puts it into the clipboard. The data following the block is pulled to the former block beginning.

Copy Block/All/Sector:

- **Normally:** Copies the current block/the entire file/the current sector into the clipboard. The contents of the clipboard can be pasted or written later.
- **As Unicode/ANSI**: Specifically copies text from the text column as UTF-16 Unicode even when the text column is not displayed in Unicode, or specifically as ANSI-encoded text even when the text column is not displayed as ANSI ASCII.
- **Into New File:** Copies the data directly into a new file (not via the clipboard). For instance, this command can be used to recover a lost file from disk sectors.
- **Hex Values:** Copies the data as concatenated hex values.
- **Editor Display:** Copies the data as text, formatted as if it was displayed in the hex editor, i.e. with an offset, a hex and a text column.
- **GREP Hex:** Copies the data as hex values in GREP syntax.
- **C/Pascal Source:** Copies the data as C/Pascal-formatted source code into the clipboard.

Paste Clipboard: Inserts the clipboard contents at the current position of a file. The file data following this position is moved forward.

Write Clipboard: Copies the clipboard contents to the current file at the current position. The data at this position is overwritten. If the end of the file is encountered, the file size is increased so that the clipboard contents finds place.

Paste Clipboard Into New File: Creates a new file of the clipboard contents.

Empty Clipboard: This command is used to free the memory used by the clipboard.

Remove: Deletes the current block from the file. The data following the block is pulled to the former block beginning. The clipboard is not affected by this command. If the block is equally

defined in all open files (i.e. it begins and ends at the same offsets), this command can even be applied to all open files at the same time.

Paste Zero Bytes: Use this command to insert zero bytes at the current position of a file.

Add Block as Virtual File: (forensic license only) If you manually define a block in Volume/Partition/Disk/File mode, this command allows you to add it to the volume snapshot as a carved file, or (in case of File mode) as a child object of the original file. Useful if you wish to treat data in a certain area (e. g. HTML code or e-mail messages found floating around in free space) as a file, e.g. to view it, search it specifically, comment on it, add it to a report, etc. If you manually carve a file within another file in File mode, the resulting file will be marked in the Attr. column as an excerpt and can be filtered as such. Already carved areas in host files are highlighted in File mode. Useful to remind the user whether he or she already has created excerpts from a file and where (e.g. from a large free space virtual file) when continuing to look at that host file.

Define Block: This function is accessible from the menu and the status bar. A dialog box lets you specify the desired block limits. This command can also be applied to all open files.

Select All: Defines the beginning and the end of the current file as its block limits.

Superimpose Sectors: see below

Convert: cf. Conversions

Modify Data: see below

Fill Block/File/Disk Sectors: see below (Wiping and Initializing)

4.5 Search Menu

Simultaneous Search: see above

Indexing, Search in Index: see above

Optimize Index: see above

Export Word List: Available once an index has been created. Allows to save a list of all the word in the index to a text file. In that list, each word that occurs in the files that were indexed will be present, and only contained once. Useful for a customized dictionary attack.

Find Text: This command is used to search for a specified string of up to 50 ASCII characters in the current file, disk or RAM section (cf. Search Options). Only supports those Unicode characters that are in the 0x00...0xFF range. For a more powerful search variant try Simultaneous Search.

Find Hex Values: This command is used to search for a sequence of up to 50 two-character hex values (cf. Search Options).

Replace Text: Use this command to replace occurrences of a specified string with another string (each of up to 50 ASCII characters), cf. Replace Options. Only supports those Unicode characters that are in the 0x00...0xFF range.

Replace Hex Values: Functions exactly as the Replace Text command, but is applied to a sequence of hex values (50 at max.), cf. Replace Options.

Combined Search: Provides a complex search mechanism. In the current and in a second file a common offset is searched, where each file contains the specified respective hex values.

Integer Value: Enter an integer (within the limits of the signed 64-bit integer data type). This function searches data in the current file, which can be interpreted as this integer.

Floating-Point Value: Enter a floating-point number (e.g. 12.34 = 0.1234 * 10^2 = 0.1234E2) and select a floating-point data type. This function searches data in the current file, which can be interpreted as this floating-point value.

Text Passages: Use this command to look for a sequence of letters (a-z, A-Z), digits (0-9) and/or punctuation marks. It is useful for instance if you intend to translate text passages hidden somewhere in a file with executable code.
Set the sensitivity of the search by specifying how long a character sequence must be to be recognized. Click “Tolerate Unicode characters” in order to force the algorithm to accept zero bytes between two characters.

Continue Global Search: This command is used to continue a global search operation (i.e. a search operation applied to all opened files) in the next file.

Continue Search: Lets you continue a search operation in the current file at the current position.

4.6 Navigation Menu

Go To Offset: Moves the current position to the specified offset. Normally this is done relative to the beginning of the file (offset 0). You can also move the cursor relative to the current position (forward or backward) or from the end of the file (backward). An offset can be specified in bytes (default), words (2 bytes), doublewords (4 bytes), records (if defined), or sectors. Press **F11** to repeat the last position movement.

Go To Page/Sector: Browses to the specified page, sector, or cluster. Note that the data area on FAT drives starts with cluster #2. The Go To Sector dialog, when applied to a physical disk, optionally allows to jump to the designated sector within the respective partition window, so that you can immediately see the allocation status of the corresponding cluster. Only for ordinary partitions, not Windows dynamic volumes or LVM2 volumes.

Go To FAT Entry/FILE Record: Jump to a certain entry in the file allocation table on a FAT drive or to a certain FILE record in the master file table on an NTFS drive, respectively.

Move Block: Moves the current block *selection* (not the data within the block) forward or backward. Specify the distance in bytes. Press **ALT**+**F11** to repeat the last block movement, press **SHIFT**+**ALT**+**F11** to reverse the movement. This command may facilitate editing a file that consists of homogeneous records of a fixed length.

WinHex and X-Ways Forensics keep a history of your offset jumps within a file or disk and allow to go **back** and **forward** in the chain later. Forensic license only: With Back and Forward you can also conveniently go back to a certain directory browser setting. This takes into account: explored path, recursive or non-recursive, sort criteria, on/off state of all filters, settings of some of the filters, some directory browser options. The Back and Forward commands also allow to activate the previously active data window again when switching between windows.

Go To...

Beginning Of File: Display the first page of the current file and moves the current position to offset 0.

End Of File: Displays the last page of the current file and moves the current position to the last byte (offset = file size - 1).

Beginning Of Block: Moves the current position to the beginning of the current block.

End Of Block: Moves the current position to the end of the current block.

Mark Position: Marks the current position and thus enables you to find it again later.

Delete Marker: Removes the marker from the screen.

Go To Marker: Moves the current position to the marker set by Mark Position.

Position Manager: see below

4.7 View Menu

Text Display Only: Hides the hex column and uses the full width of the editor window for the text display.

Character Set: Allows you to choose from ANSI ASCII, IBM ASCII, any other code page, and the Unicode character set for the text column. Keyboard input is supported only for ANSI and IBM ASCII. You may also use **SHIFT**+**F7** to change the active character set. It makes sense to select "IBM ASCII" only when viewing or editing files belonging to a DOS program. Select Unicode if you wish to read text in a language that is not Western European if it is stored in UTF-16. Unicode characters are always expected at even offsets. Select a specific code page if

necessary. The default setting is ANSI ASCII. It uses the most efficient and uncomplicated display method in WinHex, invoking only the most simple Windows API functions, and it seems to always show character interpretations according to code page 1252, even if regional settings in Windows are different, if in the font selection dialog (accessible via General Options) the "Western" script is selected.

Hex Display Only: Hides the text column and uses the full width of the editor window for the hexadecimal data display.

Record Presentation: When editing subsequent data records of the same size (for instance, table entries of a database) you may now have WinHex display every other record with a different background color, as a kind of visual aid. The color can be selected in the General Options dialog. Also, WinHex offers to display the current record number and the offset within that record (relative offset) in the status bar, based the record size and the offset of the first record as specified.
If any of the two record features is enabled, the Go To Offset command allows moving the current position in units of the current record size. If relative offsets are enabled, the Page Dn/Up keys move the cursor in units of the record size, except if you hold the Ctrl key.

Show: The **Case Data** window is part of the forensic user interface of WinHex/X-Ways Forensics and required for working with a case (when hiding the window, the case is closed). The **directory browser** is available for logical drives/partitions opened with the disk editor. The **Data Interpreter** is a small window that provides "translation services" for the data at the current cursor position. The **toolbar** is displayed optionally, too. A **tab control** makes each edit window accessible with a single mouse click only. The **info pane** provides in-depth information on any open object (file, disk, RAM).

Template Manager

Tables: Provides four conversion tables (cf. ANSI ASCII/IBM ASCII).

Lines & Columns

Synchronize Scrolling: Synchronizes up to four tiled windows on identical absolute offsets. Hold the Shift key when enabling this feature to tile the windows horizontally instead of vertically.

Synchronize & Compare: Synchronizes up to four windows and visually displays byte value differences. If no more than two windows are involved, WinHex maintains the initial distance between the offsets of the first shown byte in these windows when scrolling. Not synchronizing on absolute offsets is useful for example when comparing two copies of the file allocation table, which are obviously at different offsets. You may jump to the next or to the previous byte value difference by clicking the extra arrow buttons that are provided in one of the two edit windows.

Refresh View: Redraws the contents of the current edit window. In case the current file was updated by an external program, WinHex offers to dismiss any changes made in WinHex and reload the file from scratch.

Also refills the directory browser if the directory browser has the input focus. Useful for example when a filter for tagged items is active and you remove the tag marks of some of the listed files, if you wish to update the listing in the directory browser and get rid of those files that are no longer tagged.

4.8 Tools Menu

Open Disk: See chapter “Disk Editor”.

Clone Disk: See chapter “Disk Cloning”.

Explore recursively: Changes into a recursive view for the directory that is currently listed in the directory browser or back to the normal view. A recursive view means that not only files will be listed that are contained directly in the current directory, but also all files in all subdirectories of that directory and their subdirectories etc. For example, this allows to copy/recover selected files from different paths in a single step.

File Recovery by Type: See below.

Take New Volume Snapshot: Available for partitions with one of the supported file systems. WinHex traverses all cluster chains and thereby generates a drive map. This enables WinHex to fill the directory browser and to display for each sector which file or directory it is allocated to. It is recommended to invoke this command again after file operations on a drive to keep the information displayed by WinHex up to date. Cf. Security options.

Initialize Free Space: Confidential information is possibly stored in currently unused parts of a drive as a result of normal delete, copy and save actions. Free space on a drive can be initialized for security reasons. This effectively overwrites all data in unused parts of the disk and makes it impossible to recover this data. Available for partitions opened as drive letters. *Available in WinHex only, not in X-Ways Forensics.*

Initialize Slack Space: Overwrites slack space (the unused bytes in the respective last clusters of all cluster chains, beyond the actual end of a file) with zero bytes. This may be used in addition to "Initialize Free Space" to securely wipe confidential data on a drive or to minimize the space a compressed disk backup (like a WinHex backup) requires. Close any running or resident program that may write to the disk prior to using this command. *Available in WinHex only, not in X-Ways Forensics.*

Initialize MFT Records: On NTFS volumes, WinHex can clear all currently unused $MFT (Master File Table) FILE records, which may contain metadata (e.g. names) and even contents of previously existing files. *Available in WinHex only, not in X-Ways Forensics.*

Initialize Directory Entries: On FAT volumes, WinHex can clear all currently unused directory entries, to thoroughly remove traces of previously existing files or earlier names/locations of existing files from the file system. Useful especially in conjunction with the function to initialize all free space. *Available in WinHex only, not in X-Ways Forensics.*

Scan For Lost Partitions: Formerly existing hard disk partitions that were not automatically found when opening a physical hard disk (or an image of a physical hard disk) may be found and properly identified with this command. This command searches for the signature of master boot records, partition table sectors, FAT and NTFS boot sectors via the 0x55 0xAA signature plus for Ext2/Ext3/Ext4 superblocks, optionally only from the first sector that follows the last (location-wise) partition that was already found, and lists newly found partitions in the directory browser. Works with sector size 512 bytes only.

Interpret as Partition Start: When you find the start sector of a volume (e.g. lost partition) on a physical disk, this menu command allows you to make such a partition easily accessible via the Access button menu. If no known file system is detected starting at the currently displayed sector, you will be asked for the number of sectors that you wish to include in the newly defined partition.

Set Disk Parameters: Using this command on a physical disk, you may override the total number of sectors or optionally (can be left blank) the number of cylinders, heads, and sectors per track (all practically meaningless nowadays). This might be useful to access surplus sectors at the end of the disk (in case the total number of accessible sectors was not detected correctly), or to adjust the CHS coordinate system to your needs. Alternatively, you have the option to change the detected sector size of a physical hard disk or image, as used internally in the program for various navigation and computation work. If you should adjust the sector size, the sector count is adjusted accordingly. For example, if you change the detected sector size from 512 bytes to 4 KB (i.e. you multiply it by 8), then the total number of sectors is automatically divided by 8 to keep the same total detected disk capacity (assuming the capacity was detected correctly).

Open Memory: See chapter “Memory Editor”.

View: Available only with a forensic license. Invokes the internal viewer.

External Viewer: Invokes external file viewing programs such as Quick View Plus etc., as selected in the Options menu, and opens the current file.

Invoke X-Ways Trace: Available only if X-Ways Trace is installed. This software can analyze the history/cache files of various Internet browsers.

Calculator: Runs the Windows calculator “calc.exe”. Switching to scientific mode is highly recommended.

Hex Converter: Enables you to convert hexadecimal numbers into decimal numbers and vice versa. Simply type in the number and press **ENTER**.

Tables: Provides four conversion tables (cf. ANSI-/IBM-ASCII).

Analyze Block/File: Scans the data within the current block/the entire file and counts the occurrences of each byte value (0...255). The result is graphically displayed by proportional vertical lines. The number of occurrences and the percentage are displayed for each byte value

when moving the mouse over the corresponding vertical line.
Use this command for instance to identify data of unknown type. Audio data, compressed data, executable code etc. produce characteristic graphics. Use the context menu of the window to switch zero byte consideration on or off, to print the analysis window, or to export the analysis to a text file.
When analyzing small amounts of data (<50,000 bytes), the compression ratio that zlib achieves for that data is displayed in the analysis window caption, which also allows to draw conclusions about the nature of the data.

Compute Hash: Calculates one of the following checksums/digest of the entire current file, disks, or the currently selected block: 8-bit, 16-bit, 32-bit, 64-bit checksum, CRC16, CRC32, MD5, SHA-1, SHA-256, or PSCHF.

4.9 File Tools

Concatenate: Select several source files that are to be copied into one destination file. The source files are not affected.

Split: This command creates several destination files using the contents of a single source file. Specify a split offset for each destination file. The source file is not affected by this function.

Unify: Select two source files and one destination file. The bytes/words from the source files will be written alternately into the destination file. The first byte/word originates from the source file that was specified first. Use this function to create a file with odd and even bytes/words originating from separate files (e.g. in EPROM programming).

Dissect: Select a source file and two destination files. The bytes/words from the source files will be written alternately into the destination files. The first byte/word will be transfered to the destination file that was specified first. Use this function to create two separate files each containing either the odd or the even bytes/words of the original file (e.g. in EPROM programming).

Compare: This command is used to compare two edit windows (files or disks) byte by byte. Decide whether different or identical bytes shall be reported. You may indicate how many bytes to compare. If desired, the operation can abort automatically after having found a certain number of differences or identical bytes. The report is stored as a text file, whose size might otherwise grow dramatically.
The comparison starts at the respective offsets specified for each edit window. These offsets may differ, such that e.g. the byte at offset 0 in file A is compared to the byte at offset 32 in file B, the byte at offset 1 with the one at offset 33, etc. When you select an edit window for comparison, the current cursor position will automatically be entered in the “From offset” box.
There is yet another compare function in WinHex: you may also compare edit windows visually and synchronize scrolling in these windows (see View menu).

Create Hard Link: Cool function to create hard links of files in NTFS volumes. Useful for example to play around with hard links when attending NTFS file systems training, or if you

would like to add the same image to the same case again, which is only possible under a different name, or if you would like to create a hard link to xwforensics.exe named WinHex.exe, in order to run X-Ways Forensics as WinHe. First you select the existing file, then a path and name for the additional hard link.

Copy Sparse: Can copy a selected file and preserves the sparse nature if it is an NTFS sparse file, in the destination file. That means for example when copying a 1 TB skeleton disk image that only has 100 MB of data allocated, the copy process will finish almost instantly because only 100 MB out of 1 TB of data have to be copied. Conventional copy functions do not preserve the sparse nature of a file and copy the amount of data as indicated by the nominal file size, even if most of the data is internally unallocated and read virtually as binary zeroes.

Replicate Directory: Copies a directory with all its files and subdirectories, recursively, and recreates individually NTFS-compressed source files as NTFS-compressed in the respective output folder if supported by the destination file system and any layer in between. The command does not retroactively compress such files after their creation, but writes them immediately as compressed, which is more efficient. However, it still has to copy/send the decompressed amount of data of the source file. Supports overlong paths. Select the source directory first, then specify/create the destination directory. This function is useful for example if you wish to copy or move a case directory, which contains a few NTFS-compressed files that would be inefficient to store as uncompressed. Note that alternatively you can open a case and use the Save As command in the Case Data window for the same effect.

Wipe Securely: This command is used to erase the contents of one or more files irrevocably on magnetic disks, such that they cannot be restored by WinHex or other special data recover software. Each selected file is overwritten with data as selected by the user, shortened to a length of zero and then deleted. The name entry of the file is erased as well. Even professional attempts to restore the file will be futile. Therefore this command should be applied to files with confidential contents that are to be destroyed. *Available in WinHex only, not in X-Ways Forensics.*

Delete Recursively: This command can be used to recursively delete a directory with all its subdirectories if they cannot be deleted with Windows Explorer or other Windows tools and commands because of illegal characters in the directory names or because of missing rights (for example if "Trusted Installer" is the owner) if you can get those rights (if you are running WinHex with administrator rights). Note that you cannot apply this command to such a problematic directory itself, only to a parent directory.

4.10 Specialist Menu

Specialist and forensic licenses only.

Refine Volume Snapshot: see separate chapter

Technical Details Report: Shows information about the currently active disk or file and lets you copy it e.g. into a report you are writing. Most extensive on physical hard disks, where details for

each partition and even unallocated gaps between existing partitions are pointed out. Under Windows 2000 and XP, WinHex also reports the password protection status of ATA disks.
Forensic license only: WinHex is able to detect hidden host-protected areas (HPAs, a.k.a. ATA-protected areas) and device configuration overlays (DCO areas) on IDE hard disks up under Windows 2000 and XP. A message box with a warning will be displayed in case the disk size has been artificially reduced. At any rate, the real total number of sectors according to ATA, if it can be determined, is listed in the details report. Some important SMART status information is also displayed, for hard disks connected via [S]ATA that support SMART. Useful to check for one's own hard disk as well as that of suspects. For example, you can learn how often and how long the hard disk was used and whether it has had any bad sectors (in the sense that unreliable sectors were replaced internally with spare sectors). If a hard disk is returned to a suspect and he or she consequently complains about bad sectors and accuses you of having damaged the disk, a details report created when the hard disk was initially captured can now show whether it was already in a bad shape at that time. Also, seeing that spare sectors are in use means knowing that there is additional data to gain from the hard disk (with the appropriate technical means).

Interpret Image File As Disk: Treats a currently open and active disk image file as either a logical drive or physical disk. This is useful if you wish to closely examine the file system structure of a disk image, extract files, etc. without copying it back to a disk. If interpreted as a physical disk, WinHex can access and open the partitions contained in the image individually as known from “real” physical hard disks.
WinHex is even able to interpret *spanned* raw image files, that is, image files that consist of separate segments of any size. For WinHex to detect a spanned image file, the first segment may have an arbitrary name and a non-numeric extension or the extension “.001”. The second segment must have the same base name, but the extension “.002”, the third segment “.003”, and so forth. The Create Disk Image command can image disks and produce canonically named file segments. Image segmentation is useful because the maximum file size supported in FAT32 file systems or on media such as DVD is considerably limited. Four-digit extensions of raw image file segments are also supported for interpretation.
In some rare cases WinHex may be unable to correctly determine the nature of the image, i.e. whether it is an image of a physical disk or of a volume, consequently interprets the data in the image in a wrong way. If so, hold the Shift key when invoking this command. That way WinHex will ask you and not decide on its own. That will also make WinHex prompt you for the correct sector size and in the case of raw images for an additional storage location of further image file segments (in case you had to spread them across two different drives).
Should there be any problems with detecting the file system in a volume, you may hold the Shift key when opening the volume to indicate the file system type you suppose in the volume.
Mode 1 and Mode 2 Form 1 ISO CD images with 2,352 bytes per sector are also supported, if they are not spanned, and (with a *forensic* license) also main memory dumps. Also VMware's Virtual Machine Disk images (VMDK) and dynamic Virtual PC VHD images can be interpreted. VMDK images with ESXi Host Sparse Extents (also referred to as "Copy-on-Write Disks" or COWD), as used by ESXi servers e.g. for virtual machine snapshots, are not supported. Only allocated areas in virtual machine images can be edited. With a *forensic* license, WinHex can also interpret .e01 evidence files, which can be created with the Create Disk Image command.

The Technical Details Report also checks for certain read inconsistencies that can occur with flash media (for example USB stick of certain brands/models, but not others) in data areas that have never been written/used, where the data is undefined. The data that is read in such areas, for

example when imaging the media, may depend on the amount of data that is read at a time with a single internal read command. The result is mentioned in the report. If inconsistencies are detected ("Inconsistent read results!" in the report), you will see a message box, which offers to read sectors in smaller chunks from that device as long as it is open, which likely yields the expected zero value bytes instead of some random looking non-zero pattern data when reading such areas. Use of this option does not give you data that is somehow more accurate or original (undefined is undefined and does not mean zeroed out) or contains more or less evidence, it can just have a big impact on compression ratio achieved and reproducibility of hash values with other tools, which may use different chunk sizes for reading and thus produce different data and hash values. Note that it is possible that read inconsistencies occur that are not detected by X-Ways Forensics, because a complete check would be very slow. Again, these inconsistencies are not fatal and not the fault of the software, and they can be explained. Note that the Technical Details Report is routinely created already when you start disk imaging with the File | Create Disk Image command, so you do not need to invoke the report yourself prior to imaging.

Reconstruct RAID System: see below

Gather Free Space: Traverses the currently open logical drive and gathers all unused clusters in a destination file you specify. Useful to examine data fragments from previously existing files that have not been deleted securely. Does not alter the source drive in any way. The destination file must reside on another drive.

Gather Slack Space: Collects slack space (the unused bytes in the respective last clusters of all cluster chains, beyond the actual end of a file) in a destination file. Otherwise similar to Gather Free Space. WinHex cannot access slack space of files that are compressed or encrypted at the file system level.

Gather Inter-Partition Space: Captures all space on a physical hard disk that does not belong to any partition in a destination file, for quick inspection to find out if something is hidden there or left from a prior partitioning.

Gather Text: Recognizes text according to the parameters you specify and captures all occurrences from a file, a disk, or a memory range in a file. This kind of filter is useful to considerably reduce the amount of data to handle e.g. if a computer forensics specialist is looking for leads in the form of text, such as e-mail messages, documents, etc. The target file can easily be split at a user-defined size. This function can also be applied to a file with collected slack space or free space, or to damaged files in a proprietary format than can no longer be opened by their native applications, like MS Word, to recover at least unformatted text.

Evidence File Container: see above

External Virus Check: (Forensic license only.) Sends all files or all tagged files in an evidence object's volume snapshot to an external virus scanner, optionally only files with a size below a certain threshold. Files that are locked, deleted, or renamed by the virus scanner in the output directory will be added to a report table named "Virus suspected". It is the responsibility of the user to verify that a virus scanner is active, that it watches the folder for temporary files, and that it will indeed lock, delete or rename infected files. After verifying whether the file has been

locked, deleted, or renamed externally, X-Ways Forensics deletes it itself if it still exists.

Bates-Number Files: Bates-numbers all the files within a given folder and its subfolders for discovery or evidentiary use. A constant prefix (up to 13 characters long) and a unique serial number are inserted between the filename and the extension in a way attorneys traditionally label paper documents for later accurate identification and reference.

Trusted Download: Solves a security problem. When transferring unclassified material from a classified hard disk drive to unclassified media, you need to be certain that it will have no extraneous information in any cluster or sector "overhang" spuriously copied along with the actual file, since this slack space may still contain classified material from a time when it was allocated to a different file. This command copies file in their current size, and no byte more. It does not copy entire sectors or clusters, as conventional copy commands do. Multiple files in the same folder can be copied at the same time.

Highlight Free Space/Slack Space: Displays offsets and data in softer colors (light blue and gray, respectively). Helps to easily identify these special drive areas. Works on FAT, NTFS, and Ext2/Ext3 partitions.

4.11 Options Menu

General Options: see below

Viewer Programs: see below

Undo Options: see below

Security Options: see below

Data Interpreter Options: cf. Data Interpreter

Edit Mode: Allows you to select the edit mode used in Winhex globally. (The info pane's context menu allows to select the edit mode specifically for the active edit window only.)

4.12 Window Menu

Window Manager: Displays all windows and provides “instant window switching” functionality. You may also close windows and save files.

Save Arrangement As Project: Writes the current window constellation into a project file. From the Start Center you will then be able to load the project and restore editing positions in each document at any time, to conveniently continue your work right where you left it or to begin your work in case of a recurring task.

Close All: Closes all windows and thus all open files, disks and RAM sections.

Close All Without Prompting: Closes all windows and thus all opened files and disks without giving you the opportunity to save your modifications.

Cascade/Tile: Arranges the windows in the aforementioned way.

Minimize All: Minimizes all windows.

Arrange Icons: This command arranges minimized windows.

4.13 Help Menu

Contents: Displays the contents of the program help.

Setup: Allows you switch the language of the user interface. With **Initialize** you can restore the default settings of the program. **Uninstall**: Use this command to remove WinHex from your system. This works properly even if you did not install WinHex using the setup program.

Online: Opens in your browser, if you have an Internet connection, the X-Ways web site, the support forum, the newsletter subscription page, and a page where you can check your license status, retrieve the latest download links and get upgrade offers. There is also an option to check for updates online occasionally upon start-up of the software or at any time when you like. This can report the availability of later versions or new service releases of the currently used version (not pre-release versions) and allow to start the download. Does not send any data from within the program to the Internet, for example no system or user information or dongle ID, neither directly nor encrypted nor anonymized, of course no case data, not even the currently used version number, nothing. This option is active by default only if the program determines that it is running on the user's own system (if it is executed from the C: drive or if it was installed using the setup program). The check does not occur when running the program for the first time, so that you definitely have a chance to turn off this option before anything happens. Given the fact that most systems on which X-Ways Investigator and X-Ways Forensics are run do not have an Internet connection, this option has a limited effect.

About WinHex: Displays information about WinHex (the program version, your license status, and more).

4.14 Windows Context Menu

The Windows shell displays the context menu when the user clicks an object with the right mouse button. WinHex is present in the context menu only if you enable to corresponding option (see “General Options”).

Edit with WinHex: Opens the selected file in WinHex.

Open in WinHex: Lets you open all files of the selected folder in WinHex, just like the Open Folder command of the File menu.

Edit Disk: Opens the selected disk in the disk editor of WinHex. If you hold the **SHIFT** key, instead of the selected logical drive the corresponding physical disk is opened, if any.

WinHex provides its own context menus on the status bar, the Data Interpreter, and in the Position Manager.

5 Forensic Features

5.1 Case Management

The integrated computer forensics environment in WinHex can be used with a forensic license of WinHex only. It offers complete case management, automated log and report file generation, and various additional features such as gallery view, file signature check, HPA detection, and skin color detection in pictures.

When starting up WinHex for the first time, you are asked whether to run it with the forensic interface. This means the "Case Data" window is displayed, WinHex is run in read-only mode, and you are asked to make sure the folders for temporary files and for case data are set correctly, in order to prevent WinHex from writing files to the wrong drive.

In order to work with a case, make sure the "Case Data" window is visible on the left of the main window. If not, enable View | Show | Case Data.

From the File menu, you may create a new case (start from scratch), open an existing case, close the active case, save the active case, back up the case file and the entire case folder in a ZIP archive (only for files < 4 GB), or automatically generate a case report. You may add media as evidence objects to the case, or images (files that will be interpreted like media, see Specialist menu), or memory dumps, or directories on your own computer. Adding a directory instead of a whole partition or disk can be useful if a directory or a file of interest resides on a drive with many irrelevant files, if you merely wish to view, hash, or search a few of those files, check their metadata or copy them to an evidence file container etc.

A case is stored in a .xfc file (xfc stands for X-Ways Forensics Case) and in a subfolder of the same name, just without the .xfc extension. This subfolder and its child folders are created automatically when the case is created. You may select the base folder for your cases in General Options. It is not necessary to explicitly save a case, unless you need to be sure it is saved at a given time. A case is saved automatically at latest when you close it or exit the program.

In the case properties window, you may name a case according to your own conventions (e.g.

title or number). The date and time you create a case is recorded and displayed. The internal case filename is displayed as well. You may enter a description of the case (of arbitrary length) and the examiner's name, the examiner's organization's name and address. You may enable or disable the automated log feature for the whole case. Optionally, the evidence object subfolders in the case folder are always suggested as default output folders for files recovered/copied off a file system. You may wish to disable that feature if your preference is to copy files from various evidence objects into the same output folder.

You may select up to two code pages related to the case (more precisely: related to the locale where the original media related to the case were used). These code pages are used when naming .eml files based on subject lines (.eml files extracted from e-mail archives). If both code pages are identical, that does no harm. If identical to the currently active code page in Windows, they do not have any effect. These code pages are also used to convert the filenames in zip archives to Unicode. There may be further uses in future versions.

Case files can be password-protected. This does not involve encryption and is just a kind of lock. If the password is lost by a user, case files saved by X-Ways *Investigator* can be unlocked with a super-user password if such a password had already been entered in the installation used at the time when the case file was saved (undocumented on request).

When creating a new case, you have the option to make X-Ways Forensics recognize evidence objects that are physical media (not images) by their own intrinsic properties, not by the Windows disk number. Using this option will prevent earlier versions of X-Ways Forensics from opening the case. The advantage is that you may add multiple hard disks or external USB disks or sticks to the case that are attached to the computer at different times and get the same disk number assigned by Windows. Another advantage is that if the number of the same disk as assigned by Windows changes, X-Ways Forensics will still recognize the disk. Useful especially for triage, when not working with images. Please note that X-Ways Forensics may be unable to recognize external media already known to the case if next time they are attached through a different hardware write blocker. In that situation you can still use the "Replace with new disk" command in the evidence object context menu to point X-Ways Forensics to the correct disk. Note that component disks of an internally reconstructed RAID (read disks, not images) are still remembered by the Windows disk number when re-opening a RAID that you have added to a case.

When clicking the “SIDs...” button you can see a collection of all SID/username combinations encountered in that case (gathered from SAM registry hives in all Windows installations on images/media ever added to the case). They are used by X-Ways Forensics to resolve SIDs to usernames when working with that case.

The most powerful concept in X-Ways Forensics, that allows to systematically and completely review files on computer media, is the so-called *refined volume snapshot*. It is possible to refine the standard volume snapshot for all evidence objects of a case in one step, and to search all evidence objects with volume snapshots logically with the help of the virtual global case root window. Note that it is possible to generate a flat overview of all existing and deleted files from all subdirectories on an partition or image file of a partition by recursively exploring the root directory. In order to explore a directory recursively (i.e. list its contents plus the contents of all its subdirectories plus their subdirectories), *right*-click the directory in the directory tree in the

Case Data window. In order to *tag* a directory, you can click it with the middle mouse button in the directory tree.

In order to completely *delete* a case, you need to delete its .xfc file and the corresponding directory with the same name and all its subdirectories.

Export Files for Analysis: This menu command in the Case Data window can be applied to the entire case and from there to selected evidence objects, or to the active evidence object only. It uses the interface for external analysis of files to invoke external automated analysis tools such as DoublePics.

Export subtree: This context menu command in the Case Data window allows you to export a pseudo-graphical representation of the selected subtree in a Unicode text file, which is best viewed with a fixed-width font. The exported tree reflects the current state subdirectories (expanded or collapsed). The menu command is available for evidence objects and also for directories if you hold the Ctrl key when right-clicking a directory in the case tree. Remember to fully recursively expand a portion of the tree that you want to export, you can click the root of that portion and press the asterisk (multiplication) key on the numeric keypad.

5.2 Multi-User Coordination For Large Cases

All cases created or opened with v17.5 and later offer enhanced multi-user support, where X-Ways Forensics distinguishes between different examiners working with the same case at different times or at the same time and keeps their results separate. Multi-user support is especially helpful for large cases. Cases opened with v17.5 and later cannot be opened with earlier versions. A maximum of 255 users (examiners) is supported per case. Examiners are recognized internally by their Windows user accounts.

Multiple users may open the same evidence objects in the same case simultaneously for examination. By same case we mean the same case file, not a copy, stored in a shared network location or on a terminal server. X-Ways Forensics is responsible for synchronizing report table associations, comments and additions of files to the volume snapshot, and for making users aware of access conflicts before they occur and preventing them in most situations.

All related options can be found by clicking the button labelled "Multi-user support options" in the case properties dialog window. In particular, when creating the case (and only then), you can choose to make X-Ways Forensics *not* distinguish between different users. That would be useful if you know that only you will process that case and if you wish to process it on different computers where you have Windows accounts with different SIDs, so that you will always be treated as the same user. Also useful if multiple examiners are going to process the same case at different times and wish to share all their results directly, as it was the case in X-Ways Forensics before v17.5.

Another multi-user support option coordinates certain kinds of accesses to volume snapshots (related to adding items to the snapshot as well as editing comments and metadata) *more carefully*. It may have some performance benefits if disabled. Disabling this synchronization is

recommendable only for cases that are definitely only processed by 1 user at a time.

Report table associations and comments of different examiners can optionally be distinguished, by showing the creating examiner's initials (default), or alternatively other abbreviations of their names or (if no abbreviation is specified) their complete usernames. Comments and report table associations are shared between all examiners. Examiners can choose whether or not they get to see report table associations of other users or only their own associations (or, if half checked, only their own associations plus those of unknown users). The same file can be associated with the same report table only by 1 examiner. X-Ways Forensics imports and shows newly created report table associations of simultaneous other users in shared analysis mode when re-opening an evidence object or when case auto-save interval elapses or when manually invoking the Save Case command. The option to show initials for report table associations is represented as a 3-state checkbox. If half-checked, it has an effect on the directory browser only, not for the Export List or Recover/Copy command for example and not in the case report.

X-Ways Forensics remembers the "tagged", "already viewed" and "excluded" status of files separately for each examiner. You can choose to adopt the "already viewed" status of files in volume snapshots from all other examiners when opening evidence objects. That is useful if the goal is to avoid duplicate work, if you do not wish to review files that were reviewed by any of your colleagues already. Please note that individual file statuses ("tagged", "already viewed" and "excluded") as well as search hits of other users are lost if one examiners *removes* items from the volume snapshot.

Search hits and search terms are stored on a per-user basis as well. The first examiner opening an older case with v17.5 or later will absorb the search hits and search terms that were stored in the case by v17.4 or earlier. The "Multi-user support options" dialog window contains a button that allows you to import the search hits and search terms of another user. An option is available to limit the import of another user's search hits to search hits that are marked as notable or to that user's manually defined search hits (so-called user search hits). Another option allows to *take away* the search hits from the other user when importing them. Useful if the other user is going to resume his work later and will want to import *your* search hits back when he or she is taking over again, to avoid duplications of search hits, because your search hits include his or her hits already after you have imported them.

To view *all* the results of a colleague (report table associations, search hits, tag marked, already viewed status of files, exclusion status of files), you can open the case in read-only mode as him or her. For that, try the "Options..." checkbox when opening a case. You may prevent your colleagues from opening the case in read-only mode as you.

The "Options..." checkbox allows you to open a case in any of the following three modes:

1) entire case read-only (case file and volume snapshots),
2) shared analysis mode (ability to cooperatively produce report table associations, comments, search hit hits, and virtual files; tag files; remember already viewed files, exclude files)
3) full access

If the *same* user wishes to open the same case (the same copy) in more than 1 instance of the program simultaneously, that user has two options. Either

1) in the second instance the entire case (including evidence objects) is opened as read-only, *or*
2) the user opens the case as a separate, fictitious user (called his or her "alter ego") with separate file statuses, search hits, report table associations etc. (shared use of the case and the evidence objects is coordinated by X-Ways Forensics exactly as if the alter ego was a real, different examiner, even though the username is the same).

The aforementioned "Options..." checkbox allows you at any time to open the case as your alter ego, not only when opening the same case in a second instance of the program. It also allows you to open a case in shared analysis mode if it is not open anywhere else at the moment.

Multiple users running searches, creating report table associations, entering or editing comments, editing extracted metadata, tagging files, excluding files, marking files as already viewed is all supported for the same evidence object at the same time. *Removing* items from a volume snapshot while the evidence object is open somewhere else, however, is forbidden and will be refused by the program. The goal of the multi-user coordination in v17.5 and later is to support concurrent *analysis*/*review* work by multiple examiners. *Removing* files from a volume snapshot is not considered ordinary review/analysis work. Volume snapshot refinements should be done systematically *in advance*.

The initials of the examiner who has attached files to the volume snapshot or manually carved files in v17.5 and later can be seen in square brackets next to the filename, so that it is easy to tell who has introduced such files to the case.

Technical changes to the way how multiple simultaneously users are coordinated are reserved. To be on the safe side, please make sure that simultaneous users are running the same version of the software.

Last not least v17.5 allows you to review the processing history of a case in its properties. This reveals which versions were used on it (recorded only by v17.3 SR-10 and later, v17.4 SR-4 and later and v17.5 and later) and by which users (recorded only by v17.5 and later).

You may turn *off* "Coordinate processing by simultaneous users more carefully" for some performance benefits there is only user of a case at a time.

There is an option to always suggest shared analysis mode when opening a case. That mode can be useful even for the first of many simultaneous users that open the same case because only in that mode newly created report table associations are shared out to other simultaneous users at regularly intervals (depending on the case auto-save option).

Alternative Ways of Sharing Analysis Work

Option #1: Multiple computer forensic examiners can work simultaneously with their *own* copy of the same case simultaneously (always copy both the .xfc file and the corresponding subdirectory) and exchange results with each other or reconcile all results in the main copy of the case, by exporting and importing report table associations (i.e. their categorization of all the relevant files, e-mails, etc.).

Option #2: Potentially relevant files are copied from the original evidence objects to multiple evidence file containers. The containers are examined by different investigators simultaneously in newly created cases (in X-Ways Forensics or X-Ways Investigator). They also can export their report table associations, which can then be imported back into the original case.

Both commands, the export and import of report table associations, can be found in the context menu of the case tree. Export is supported at the case and evidence object level, import at the case level. The names of the examiners/investigators could be included in the names of the report tables if in the original case it should be obvious who created which associations. Please note that you cannot import report table associations in the original case any more if you have taken a new volume snapshot or if you have removed objects from the volume snapshot in the meantime, because in that situation it is not guaranteed that the internal IDs of the file remain the same and that a reliable association is possible. The import works only if you import into the same evidence object that you export from. The same evidence object in a case in X-Ways Forensics, or a copy of the same case. It does not help if it's the same image or disk in a different case. Even if it is the same case and the disk or image was removed from the case and later added again, it will not be considered the same evidence object any more. However, you (e.g. as a user of X-Ways Investigator) can export from an evidence file container in a new case and have a user of X-Ways Forensics import the report table associations into the original evidence object in the original case, from which the files in the container originate. That is possible because the evidence file container has information that allow to identify the original evidence object.

Distributed Volume Snapshot Refinement

X-Ways Forensics allows to refine the volume snapshots of *different* evidence objects of the same case using multiple machines on the same network, simultaneously, to save time through parallelization.

Each user/computer opens the same .xfc case file (the same copy on the same computer). All participating users/computers or all except for one (the master session) have to open the case as partially read-only, i.e. only allowing for shared analysis work/distributed volume snapshot refinement. This can be done by checking the “Options…” box in the Open Case dialog window, or you will be prompted automatically when opening the case if the case if already open in another session as not read-only (i.e. in the master session). Other sessions will see the refinement results at latest when refinement has completed and when the respective evidence object is re-opened. The case does not have to be closed and re-opened.

You have the option to specifically open individual evidence objects (not the entire case) with the volume snapshot treated as read-only, using a dedicated command in the evidence object context menu in the Case Data window. Please note that this has nothing to do with how the evidence object itself (the disk or the image) is treated. X-Ways Forensics never alters data in sectors of disks or interpreted images files when opening them as evidence object. Only the volume snapshot, i.e. the database with information about all the files and directories found, is either read-only or, and that is the normal state, changeable.

5.3 Evidence Objects

You may add any currently attached computer medium (such as hard disk, memory card, USB stick, CD-ROM, DVD, ...), any image file, directory or ordinary single file to the active case. It will then be permanently associated with this case (unless you remove it from the case later), displayed in the tree-like case structure, and designated as an *evidence object* or *source of evidence*. A subfolder is created in the case folder for each evidence object, where by default files will be saved that you copy/recover from that evidence object, so it will always be obvious from which object exactly (and from which case) recovered files originate. If you wish to add more than 1 file from the same directory to the case, please add the whole directory, just exclude or remove those files that are irrelevant.

In the evidence object properties window, you may enter a title or number for that evidence object according to your own conventions. You may change the order of evidence objects in the case tree using the small arrow buttons in the upper left corner, except for "dependent" evidence objects (partitions that belong to a physical disk). The date and time it was associated with the active case is recorded and displayed. The internal designation of the evidence object is displayed as well as its original size in bytes. You may enter comments of arbitrary length that apply to the evidence objects, and a technical description of it is added by X-Ways Forensics automatically (as known from the Technical Details Report command in the Specialist menu, plus some essential information about Windows installations, if found in a partition). You may have the program calculate one or two hashes (checksum or digest) on the evidence object and verify them later, so that you can be sure that data authenticity has not been compromised in between. Hashes stored in evidence files are imported automatically when added to a case. You may disable the automated log feature for a specific evidence object if the log feature is enabled for the case as a whole.

To add images or media to a case. you can use the "Add" commands in the case data window's File menu. When adding images, you can also select that the volume snapshot of newly added evidence objects should be refined immediately. Another way how to add opened images or disks to the case is the "Add" command in the context menu of the data window's tab.

The command “Replace with New Image” in the context menu of an evidence object allows you to replace a disk that is used as an evidence object in your case with an image (useful if you first preview the disk before you acquire it, i.e. created an image of it), without losing your volume snapshot, search hits, comments, etc. Can also be used to simply tell X-Ways Forensics the new path of an image in case the image was moved or the drive letter has changed, or if the image filename was changed, or if the type of the image was changed (e.g. raw image to be replaced with a compressed and encrypted .e01 evidence file). In the case of a physical, partitioned evidence object it is recommended to apply this command to that parent object (i.e. the physical disk). The change will then automatically also be applied to the child evidence objects (i.e. partitions). If the new image is an image of a different disk or a different evidence file container or an evidence file container that has been filled further, i.e. if the volume snapshots cannot match, you will likely get a warning because the size of the new image is different from the size of the previous image. Time and again, users of X-Ways Forensics try to use this command to replace an evidence object in a case with a *different* evidence object, although that doesn't make any sense because that way the technical description, the volume snapshot, any search hits,

comments and report table associations don't fit the other evidence object. These users then typically complain that they receive an error message. The message is displayed because X-Ways Forensics usually notices based on the size that the new image is a totally different image. If you don't need evidence object A any more in your case and you need add an evidence object B, then you can simply remove A and add B. There is no alternative to that, and an alternative is neither reasonable nor required.

It is possible to open an evidence object even if the disk or image is not currently available, via a special command in the evidence object's context menu, to see at least the volume snapshot. That means you can see all the file metadata stored in the volume snapshot (filename, path, file size, timestamps, attributes, etc.), can use most filters etc., but cannot see any data in sectors and cannot open/view any files.

In the Case Root window, evidence objects can be marked as important with a yellow flag, via the context menu or by hitting the Space bar. You will see that yellow flag in the Case Data window and when selecting evidence objects, for example for recursive exploration from the Case Root or when generating a report.

5.4 Case Log

When enabled in the case and the evidence properties window, WinHex obstinately logs all activities performed when the case is open. That allows you to easily track, reproduce, and document the steps you have followed to reach a certain result, for your own information and for the court room.

The following is recorded:

- when you a select a menu item, the command title (or at least an ID), and the name of the active edit window, if not an evidence object, preceded by the keyword "Menu",
- when a message box is displayed, the message text and what button you pressed (OK, Yes, No, or Cancel), preceded by the keyword "MsgBox",
- when a small progress indicator window is displayed, its title (like "Recovering files...") and whether the operation was completed or aborted, preceded by the keyword "Operation",
- a screenshot of each displayed dialog window with all selected options, e.g. for a complex operation that follows, preceded by the window's title,*
- the extensive log produced by Clone Disk and File Recovery by Type,
- your own entries (free text) that you add with the Add Log Entry command, either to the case as a whole or to a certain evidence object.

The destination path of each file copied/recovered with the directory browser context menu, along with selected metadata of that file (e.g. original name, original path, size, timestamps, ...), is logged in a separate file “copylog.html” or “copylog.txt” in the “_log” subdirectory.

All actitivities are logged with their exact date and time, internally in FILETIME format with 100-nanosecond interval precision. Logs are by default associated with the case as a whole.

However, logs of activities that apply to a certain evidence object are directly associated with that evidence object. This determines where they appear in a report. Screenshots are saved as PNG files in the "_log" subfolder of a case folder.

*If "Include screenshots in log" in the case properties is half-checked, that means that no actual screenshots of dialog windows will be taken, just a simple ASCII representation will be stored in the log (the same that you get when via Ctrl+C). These details are included in a special way in the HTML output, so that they do not detract too much from the main log entries. Either they are output in a smaller font and gray color (if "Include screenshots in log" is fully checked) or simply as a pop-up when hovering with the mouse cursor over a space-saving placeholder rectangle, as known from Windows registry reports in X-Ways Forensics (if half checked) or not at all (if not checked). The placeholder rectangle and pop-up work best when viewed in Google Chrome, as that browser does not truncate the text if lengthy and even shows a preview of the first line in the placeholder rectangle. If you have X-Ways Forensics take conventional (real) screenshots of dialog boxes in the log, pixels with the gray background color can be changed to pure white, to save toner/ink in case you are going to print your log at some time (anyway, please think twice and save paper).

5.5 Case Report

You may create a report from the File menu of the Case Data window. The report is saved as an HTML file and can thus be displayed and opened in a variety of applications. For example, you may view it in your favorite Internet browser and open and further process it in MS Word. The application to open the report in can be specified in Options | Viewer Programs. If no such program is defined, the report file will be opened in the application that is associated with the file extension on your computer. With the Open Report command you can select any existing file and open it in the defined or associated application.

The report can consist of the following elements:

- **Basic report**: Starts with an optional header line, an optional logo, an optional preface (in which you may use HTML code), the case title and details, followed by a list of hyperlinks to the individual evidence object sections. For each evidence object, the report specifies its title, details, and technical description, your comments, your annotations. If only half checked, technical details about the evidence objects are not included in the case report, the evidence objects are merely listed.

- **Report tables**: All files in selected report tables can be output to the report, with selected metadata such as filename, path, timestamps, comments. Files can be optionally copied off the evidence objects into a subdirectory of where the report is saved. Then they will also be linked from the report. Either all files can be copied or merely pictures. By default, pictures will be displayed directly in the HTML report file and not merely linked. They are resized to the maximum dimensions you specify while retaining their aspect ratio. If you specify maximum dimensions of 0×0, then the pictures will only be linked, just as other files. If you choose to reference multiple files in the same line (to render the report more compact when printing), you will appreciate that long filenames and paths

can be artificially broken into multiple lines after a user-defined number of pixels, to make sure the width does not exceed the paper size.

There is an option to only make a copy of tagged files for inclusion in a case report instead of all or none. Useful if you wish to reference all notable files with their metadata in your report, but show only a subset of those.

Files can be output either grouped by evidence object and sorted by internal ID or in the order as they are currently listed in the case root window, where you can freely change the order thanks to up to 3 sort criteria. If no files are currently listed in the case root (because it has not been explored recursively), then the second option is grayed out. Explore the case root recursively first to make it available (right-click it). Note that if you choose the second option, files that are not listed in the case root window will not be output, even if they are part of a report table. That means that current filter settings have an effect on the generation of the report, too. If files are omitted because they are not listed in the case root window at the time of report generation, you will be notified of that in the report and in a message box.

If the box to output report tables is only half checked, then only the number of items in each report table will be reported.

Many different settings allow to tweak the report to your liking. For example, "Name output files after unique ID" will ensure filenames that are succinct, unique, trackable and reproducible, and will also ensure that if the same files is associated with multiple report tables, it will be copied to the report subdirectory only once. That saves time and drive space. "List each file only once" is a 3-state checkbox. If fully checked, no file will be referenced in the report by more than one report table. Note that you can still see all report table associations of a file when it is listed in its first report table in the report, if you output the field "Report table". If the checkbox is half-checked, that means that a file will still be referenced (listed) by additional report tables in the report if it has multiple associations, but copied only once and linked only from the first report table.

- **Case log**

By default, the report is created for the entire case. Optionally it is created for selected evidence objects only. It is relatively easy to use CSS (cascading style sheets) for case report format definitions. In addition to defining the parameters for standard HTML elements, key elements of the report are assigned "class" parameters to simplify targeting those for formatting purposes. Example style sheets are available to use as a basis for further modification. The report options allow picking or editing a CSS file as part of the reporting process. The default is "Case Report.txt". The default look from v18.0 and earlier is still available as "Case Report Classic.txt".

5.6 Report Tables

In the directory browser of an evidence object, you can associate notable files with report tables. A report table is a user-defined (virtual) list of files, especially notable files. Files associated with

report tables can then be easily included in the case report with all their metadata and even links (pictures can be included directly), and you can filter by their report table association in a recursive view in order to easily locate these files later (like bookmarking files). The filter can reference multiple report tables at the same time (with OR, AND and NOT operators) and even has an option that allows to additionally include siblings of the files of a certain report table, i.e. files in the same directory. That is useful, especially when exploring recursively and sorting by path, to check whether there are any further notable files in the neighborhood.

E.g. you could create report tables like "related to company X", "evidence against suspect A", "incriminating pictures", "unjustified expenses", "forward to investigator B", "print later", "get translated", “show to witness C” etc., and later when you are done viewing files, you can get the big picture of all relevant files by using the report table filter (e.g. "Show me all files related to company X that are also considered evidence against suspect B"). You are practically assigning files to certain custom categories defined by yourself. Also allows you to revisit files later that are still be closely examined.

Having files in a dedicated report table also allows to conveniently copy/recover them in a single step at a later point of time or get a gallery overview of these files specifically. The same file can be associated with multiple report tables. This can be done in the dialog window that appears when invoking the Report Table Association command in the directory browser context menu, for one file or several selected files at a time. This dialog window does not show the existing associations of the selected file or files (that would be quite complicated to achieve anyway for multiple selected files, instead simply look at the "Report table" column), but creates new report table associations in a convenient and user-configurable way and/or removes existing associations. The program remembers the report tables selected last for creating associations. In the same dialog window you can also create new report tables, rename or delete existing ones, and remove/override previous associations. For each report table you can specify whether you would typically like to associate only the selected file or directory to that report table and/or at the same time the selected file's parent file (if any) and/or the file's or directory's child objects and/or any identified duplicates of the selected file in any currently open evidence object (duplicates that have been identified based on hash values and marked accordingly in the Attr. column, see context menu, as well as hard links except in HFS+).

Another option allows to automatically associate siblings of selected files with report tables. Useful for example when reviewing search hits, if you find a relevant search hit in the attachment of an e-mail message and want to be sure to include other attachments of the same e-mail message in further processing, even if they do not contain search hits.

If you need to categorize a lot of files with the help of report tables, you can also use keyboard shortcuts. X-Ways Forensics automatically assigns the shortcuts Ctrl+1, Ctrl+2, ..., Ctrl+9 to your report tables. In the dialog window for report table associations you can also assign these shortcuts to report tables yourself, by simply pressing the keys while a report table is selected. Ctrl+0 removes all report table associations from a file. Alternatively you may simply press the keys in the numeric pad on your keyboard if Num Lock is active, without Ctrl. This will not be considered normal input in the directory browser although the Ctrl key is not pressed. The numpad keys may not work on all computers.

There is an option to create report table associations for files based on search terms that they

contain according to the "Search terms" column. Useful if you wish to keep the information about which file contains which search terms even after deleting search hits, or to preserve it in evidence file containers. Report tables representing contained search terms are the 3rd kind of report tables, the first two being report tables created by X-Ways Forensics to make the user aware of certain file specialities and user-created general purpose report tables. Report tables representing search terms are recognized as such in evidence file containers by v17.3 and later.

It is possible to save and load lists of report table names in the report table association dialog window. This is useful to start right away with a set of predefined report tables as typically needed for a certain kind of case. The maximum number of report tables in a case is 256.

Report table associations can be exported and imported. See Alternative Ways of Sharing Analysis Work.

In order to output report tables to a report (the original purpose of report tables, hence their name), use the Create Report command in the Case Data window.

Report table associations are also used internally and created automatically by X-Ways Forensics, to make the user aware of various potential specialties of certain files. It is up to you whether you wish to follow up and take a closer look at those files or not. The names of internally created report tables are displayed as indented and in a different color, to avoid mix-up with your own report tables. Automatically generated report tables include:

No detectable textual contents
Unable to decode text
For error messages see Metadata
Unable to explore
Empty archive?
Spanned archive
No e-mails found
Path too long.
Large non-resident $EA
Animated GIF
Multi-page TIFF
Multi-page JPEG marker
Zip bomb? Not fully processed
Unexpected tail (SFX?) / Contains unknown segment (SFX?)
FSG Packer / PECompact / UPX / Unknown segment / Binder?
Contains embedded document(s)
Contains embedded object(s)
Contains embedded file
Contains hidden file
Hybrid MS Office document!
RAR hybrid
Contains embedded non-JPEG/non-PNG picture
Contains invisible old revisions
Concatenated-PDF
Contains private chunk

No pictures extracted
Reason for crash?
Unsupported file type variant
Omitted
Not copied
Virus suspected
Unable to read
Not decompressed

5.7 Internal Viewer

The internal viewer can be invoked with the “View” command in the Tools menu and in the directory browser's context menu, plus in Preview mode. It shows picture files of various file formats (JPEG, PNG, GIF, TIFF, BMP, PSD, HDR, PSP, SGI, PCX, CUT, PNM/PBM/PGM/PPM, ICO, using an internal graphics viewing library) plus the structure of Windows registry files, Windows Event Logs (.evt and .evtx), Windows shortcut files (.lnk), Windows Prefetch files, $LogFiles, $UsnJrnl:$J, Ext3/Ext4 .journal, .ds_store, Windows Task Scheduler (.job), $EFS LUS, INFO2, Restore Point change.log.1, wtmp and utmp log-in records, MacOS X kcpassword, MacOS X finder bookmarks (flnk), AOL PFC, Outlook NK2 auto-complete files, Outlook WAB address books, Internet Explorer travellog files (a.k.a. RecoveryStore), Skype Chat Sync, MS Outlook Express DBX and many other files internally. If you try to view a file that is not supported by the internal viewer, the separate viewer component is invoked instead.

There is an additional separate viewer component that integrates seamlessly and allows to conveniently view more than 270 (!) file formats (such as MS Word, Excel, PowerPoint, Access, Works, Outlook; HTML, PDF, StarOffice, OpenOffice, ...) directly in WinHex and X-Ways Forensics. This component is provided to all owners of forensic licenses issued for v12.05 and later. It can be enabled in Options | Viewer Programs, optionally also for pictures that could be displayed by the internal graphics viewer library. More information online. The folder for temporary files used by the separate viewer component is controlled by WinHex/X-Ways Forensics, i.e. set to the one the user specifies in General Options. However, unlike X-Ways Forensics, the viewer component does not silently accept unsuitable paths on read-only media. Please note that the viewer component since its version 8.2 creates files in the Windows profile of the currently logged on user, in which it stores its configuration and settings. In earlier versions, if actually used, not when merely loaded, it left behind entries in the system registry.

Registry Viewer

MS Windows maintains an internal database called registry which contains all important settings for the local system and installed software in a tree-like structure. The data is persistently stored in files called registry hives. You can open and view hives by double-clicking them in the directory browser or using the context menu. This will open them in the integrated registry viewer. Supported formats are NT/2K/XP/Va/7 hives. Win9x and WinMe hives can only be loaded by the registry viewer of X-Ways Forensics 15.9 and earlier. NT/2K/XP/Va/7 hives are located in the file “ntuser.dat” in a user profile and in the directory \system32\config.

Up to 32 hives can be opened in the registry viewer at the same time. The registry viewer has the ability to find deleted keys and values in hives that contain unused space and lost keys/values in damaged/incomplete hives. If no complete path is known for keys, they will be listed as children of a virtual key called "Path unknown".

With a right-click a pop-up menu can be opened anywhere in the window, which lets you invoke the commands “Search” and “Continue Search”. Clicking “Search” invokes a dialog that lets you specify a search expression and where you want to search. You can browse either keys or names or values or all of them. The search always starts at the topmost root of the first loaded hive and spans all opened hives. "Continue Search" finds the next match after at least one match has been found. The currently selected element is not relevant for where the search continues. The "search whole word only" option is not guaranteed to work for values.

In the right-hand window the pop-up menu also contains the command "Copy" which lets you copy the value of the selected element to the clipboard.

When clicking a value of a loaded hive in the Registry Viewer, if the data window with the drive/image from which the hive was loaded is in File mode, the cursor will automatically jump to the selected value in the registry file, and the value will automatically be selected as a block in that file. Useful as that allows to see the value in hexadecimal and text and as that allows to easily copy binary values in either binary or as text, not only as hex ASCII.

The Export List command in the registry viewer context menu allows to export all values in the selected hive to a tab-delimited text file.

When selecting a value, an edit window in the lower right corner tells you the logical size of that value and the size of its slack. It also interprets registry values of the following types, as known from the registry report: MRUListEx, BagMRU, ItemPos, ItemOrder, Order (menu), ViewView2, SlowInfoCache, IconStreams (Tray notifications), UserAssist, Timestamps (FILETIME, Epoch, Epoch8), MountedDevices, OpenSavePidlMRU, and LastVisitedPidlMRU. The edit window also displays the access rights/permissions of the registry keys if (Default) is selected.

$LogFile Viewer

Basic Concepts:
Each statement falls into one of the three categories:
1) Log-Operation
The on-disk data at (LCN,Byte offset) is to be replaced in case of a Redo/Undo-Operation with the one specified within the log operation.
2) The PAGE statement indicates the start of a new log page (multiple of 4 KB). The LSN specifies the last end LSN for this page. A * marks a stale page.
3) The CheckPoint statement specifies a LSN to restart with.
Each statement is preceeded by an byte offset pointing into the $LogFile.

Abbreviations:
LSN=Logical Sequence Number

LCN=Logical Cluster Number
VCN=Virtual Cluster Number
FID=File ID

Limitations:
Only log operations are shown which affect on-disk structures. FILE records and INDX buffers are not completely dumped. For complete data, follow the byte offset displayed for the operation of interest. An NTFS journal is only processed if the path of such a file contain the string $LogFile.

5.8 Registry Report

From within the registry viewer, WinHex can create an HTML report, listing values of possibly relevant registry keys, when you invoke the command "Create Registry Report" in the right-click pop-up menu. The registry keys that are to be reported in all open hives are defined in text files like the pre-supplied “Reg Report *.txt”, which can be tailored to your needs. The registry files you view must have their original names, or else the report may fail. You may edit the list of registry keys in this files to tailor the report to your own needs.

Standard tables have 4 columns: description, extracted value, registry path (provided as a tooltip), and last modification date of the corresponding key. The dates are displayed in gray for values that are not the only values in their respective key, as a visual aid to remind the reader that they are not the modification dates of the values themselves.

Free space in registry hives can be analyzed with the report definition file "Reg Report Free Space.txt". The free space can be as large as several MB, especially as a consequence of the use of virus scanners and registry cleaning programs. Deleted registry values are now highlighted in the report in red color.

Also registry value slack has a relevant size in NTUSER.DAT hives. This fact is exploited with 2 measures:
1) If the slack contains text strings, it will be output in the registry report (in green). This new feature can optionally be turned off the registry viewer context menu.
2) For values that contain item lists (i.e. are binary) you can use the "Reg Report Free Space.txt" definitions to output registry report will output lists of filenames with timestamps in green. The first timestamps is an access date, the second one is a creation date. If no timestamps can be output, these are artifacts from "RecentDocs".

Format of entries in "Reg Report *.txt"

(*type*) (*tab*) (*registry path*) (*tab*) (*description*) (*linefeed*)

type:
?? definition for any Windows version
NT for Windows NT through XP
VT for Windows Vista and 7

** new function (without absolute paths)
FR query in free space of the hive

registry path:
Full path of registry keys

HKLM: HKEY_LOCAL_MACHINE
HKCU: HKEY_CURRENT_USER

If an asterisk ("*") is provided as the last key, all keys on the same level and deeper and their values will be included in the report.

example:
NT HKLM\Software\Microsoft\Windows\CurrentVersion* report whole Windows branch

If you wish to report a particular value that exists in all subkeys of a certain key, you can as well write an "*" for all subkeys and include the value after that.

The generated report contains the registry path with its timestamp, the filename of the registry hive that the key was found in, the description that was provided in the "Reg Report *.txt" file, and the value.

The description field may contain an additional statement at the end that starts with a % character. If the % is followed by a numeric character n, the n-th element of the registry path will be appended to the description in the report. This can be very useful if the path and not the value (or not only the value) contains the relevant information. If the % is followed by a letter, the value will be preferably interpreted as the data type that the letter stands for. The following letters and data types are defined at the moment:
%f Windows FILETIME timestamp
%e Epoch (Unix) timestamp
%E Epoch8 (Unix) timestamp as QWORD.
%T Windows system time timestamp
%s ANSI-ASCII null-terminated
%S UTF-16 string null-terminated
%b binary data not to be interpreted as characters (REG_BINARY)
%P Windows PIDL data structure
%I ItemPos data structure (covers Shell Bag, desktop shortcuts, and more)
%B conditional: if value TRUE
%F conditional: if value FALSE
%- no empty mode
%+ recursion of the subtree
%i value case-insensitive
%d deleted values only
It is also possible to combine numeric characters and letters (e.g. %10f). In that case the numeric character must precede the letter.

// at the start of a line comments out that line (will cause it to be ignored).
at the start of a line will output explanatory text into the report.

Additional output

In a second phase of the creation of the registry report, additional data will be analyzed and output as tables at the end of the HTML file. The specifications in the definition file which belong to this second phase are marked with "Dummy". This causes the first phase to prevent any normal output. If you would like to get the output of the first phase, you merely need to change the description in the definition to anything other than "Dummy".

The table "Attached devices by serial number" is created according to the algorithm that Harlan Carvey describes in chapter 4 of his book. Furthermore you can find the tables "Partitions by disk signature", "Windows portable devices", "Drivers installed", "File systems installed", "Services installed", "Networks", and "Network cards".

Another table is called "Browser Helper Objects", compiled with data from the hives NTUSER.DAT and SOFTWARE, about browser usage. "External Memory Devices" is a table which can be retrieved from Software hives of Windows Vista and later that lists external media with access timestamps, hardware serial number, volume label, volume serial number and volume size (size often only under Vista). Select the definition file "Reg Report Devices.txt" to get the table.

5.9 Simultaneous Search

This search command in the Search menu is available for owners of specialist and forensic licenses, and offers all options only for owners of forensic licenses. This search is simultaneous in that it allows the user to specify a virtually unlimited list of search terms, one per line. The occurrences of these search terms can be saved and listed in an evidence object's search hit list (forensic licenses, when working with a case), or in the general Position Manager.

You may use the simultaneous search to systematically search multiple hard disks or disk images in a single pass for words like “drug”, “cocaine”, (street synonym #1 for cocaine), (street synonym #2 for cocaine), (street synonym #3 for cocaine), (street synonym #3 for cocaine, alternative spelling), (name of dealer #1), (name of dealer #2), (name of dealer #3), etc. at the same time. The search results can narrow down the examination to a list of files upon which to focus.

The simultaneous search can be used to search physically in sectors or logically in file or in a previously created index. Physically, it searches the sectors on a medium in LBA order (except if you search upwards, then in reverse order). If you do not have WinHex list the hits of a physical search, you may use the F3 key to search for the next hit. Logically, the search proceeds file by file, which is preferable and much more powerful and thorough. More about the logical search.

You can search the same search terms simultaneously in in up to 6 code pages. The default code page, that is active in your Windows system, is marked with an asterisk and initially preselected. E.g. on computers in the US and in Western Europe, the usual default code page is 1252 ANSI Latin I. The code pages named "ANSI" are used in Microsoft Windows. "MAC" indicates an

Apple Macintosh code page. "OEM" indicates a code page used in MS-DOS and Windows command prompts. If a search term cannot be converted to the specified code page because of characters unknown in that code page, a warning is issued. Code page independent GREP searches for exact byte values are possible when searching in a "non" code page called "Direct byte-wise translation for GREP", which translates byte values without any mapping for certain code pages or case matching. X-Ways Forensics also allows to search in both little-endian and big-endian UTF-16, and in any regional Windows code page plus UTF16 with the MS Outlook cipher (compressible encryption) applied.

You can define which characters should be considered to be parts of words. This is useful to avoid false hits for short real language words in binary garbage data or Base64 code and generally for users that consider numbers to be parts of words (such as in “GIF89”). Example: An undesirable hit for "band" in "7HZsIF9BAND4TpkSbSBS" can be prevented if you search for it as a whole word only if you redefine the alphabet to include digits 0-9, i.e. consider them word characters.

It is possible to review the (incomplete) search hit list in the middle of an ongoing simultaneous search. You can click the search hit list button at any time to view the preliminary search hit list. Additional search hits that have been collected as the search continues will be listed when you refresh the search hit list, by clicking the Enter button in the search term list as usually. This approach to view preliminary search hits is useful e.g. when previewing a live system on site to determine whether a medium might contain relevant files and should be captured. If after searching 5% of the data and reviewing the search hits gathered so far the answer is Yes, the search can be stopped already and a lot of time is saved.

5.10 Logical Search

Powerful subvariant of the simultaneous search. Allows to search either all files, all tagged files, or (if invoked from the directory browser context menu) all selected files. File slack can be specifically included or excluded. The logical search has several advantages over a physical search:

- The search scope can be limited to certain files and folders, through tagging or selecting files. Please note that the amount of data to search that may be displayed in the dialog window is an estimate only. The actual scope of the search may vary because of slack space.

- Searching in files (usually = in the cluster chains allocated to files) will find search hits even if the search term happens to be physically split in a fragmented file (occurs at the end and the beginning of discontiguous clusters).

- A logical search can be successful even in files that are compressed at the NTFS file system level, as they are decompressed for searching. This holds true even for files that were found via a file header signature search, if that was specially adapted for NTFS compression.

- If the contents of archives (files in ZIP, RAR, GZ, TAR, BZ2, 7Z, and ARJ, if not encrypted, forensic license only) and individual e-mail messages and attachments have been included in the volume snapshot, they can be searched as well.

- The text that is contained in files whose format is supported by the viewer componet, e.g. PDF (Adobe), WPD (Corel WordPerfect), VSD (Visio), SWF (Shockwave Flash), can automatically be extracted/decoded/decompressed prior to search, resulting in unformatted ASCII or UTF-16 plaintext, which can be reliably searched in addition to the original data itself. Search hits might otherwise be missed because various file types typically or at least sometimes store text in an encoded, encrypted, compressed, fragmented or otherwise garbled way. Important: In particular for HTML, XML and RTF documents as well as HTML-formatted e-mail messages in .eml files, which may employ various methods of encoding (e.g. UTF-8) non-7-bit-ASCII characters (e.g. German umlauts), decoding may be useful, depending on the language of your search terms/the characters contained in your search terms. When you specify a file mask for decoding, that mask will not only be applied to the names of searched files, but also to their true type if verified by signature (see Refined Volume Snapshots). This feature requires the separate viewer component to be active for the decoding and text extraction part. The decoded text is output in Latin 1 or Unicode, and can optionally be buffered (cf. Options | Viewer Programs) to allow for a convenient context preview for search hits in the decoded text and to accelerate future searches. The default file mask for this option is *.pdf;*.docx;*.pptx;*.xlsx;*.odt;*.odp;*.ods;*.pages;*.key;*.numbers;*.eml;*.wpd;*.vsd. It is recommended to add ;*.html;*.xml;*.rtf depending on the characters searched for, and more depending on your requirements. For example *.doc might be a good idea if you want to be very thorough because text can be fragmented or change from one character set to another abruptly in the middle of a MS Word document. Just keep in mind that the additional decoding and search require more time and like result in duplicated search hits (search hits found in both the original format and the result of the text extraction). E-mails will generally not be decoded by X-Ways Forensics when only 7-bit ASCII characters are search. The file mask is applied to both the filename and the detected true file type. To see what text is extracted from a document by this function, you can select the document in the directory browser in Preview mode and hold the Shift key when switching to Raw mode.

- If you are not interested in each and every search hit, but merely in which files contain at least one the specified searm terms, a logical search can be greatly accelerated by telling X-Ways Forensics that only one hit per file is needed, so that it can skip the remainder of a file once a hit has been recorded and continue with the next file. The resulting search hit list will be inherently and systematically incomplete, and no assumption must be made that somehow “the most useful” search hit in each file will be collected, or, if multiple search terms are used, a search hit for a search term that you consider more important will be collected. However, it is guaranteed that it contains all the files for which there was at least one hit (for one of the search terms used), and each such file once only. Such a list is sufficient (and efficient!) to manually review the affected files, comment on them, copy the files off an image or pass them on to other investigators in an evidence file container etc. Note that of course it is not possible to combine search terms with a logical AND if only 1 hit per file was recorded. That consequence is typically forgotten by unsuspecting

users.

- Files that have been marked as irrelevant by hash computation and hash database matching or files that have been excluded by the user or that are filtered out by an active filter can be omitted from a logical search to save time and reduce the number of irrelevant search hits. The slack of such files is still covered if the option "Open and search files incl. slack" is fully checked, so that this option has a higher priority. If only half checked, the slack of such files is omitted, too.

- The recommendable data reduction specifically omits certain files from the search to avoid that time is wasted or duplicate hits are produced unnecessarily.

  E-mail archives of the types MBOX and DBX as well as file archives of the supported types (ZIP, RAR etc.) will not be searched if the e-mails and files that they contain have already been included in the volume snapshot, in order to save time. In that case *only* those extracted e-mails and files will be searched, in their natural (unencoded and uncompressed) state. This may be reasonable for keyword searches and in particular for indexing (which has a hard time processing e.g. Base64 code), but not necessarily for technical searches for signatures etc. Using this option constitutes a compromise. The slack of archive files is still included if the file slack option is enabled, as that option has a higher priority.

  A file that that is marked as renamed/moved will not be searched either if data reduction is enabled and if principally all files in the volume are to be searched (as opposed to tagged or selected files only) because the same file will already be searched under its current name/in its current location.

  If *.docx;*.pptx;*.xlsx;*.odt;*.odp;*.ods;*.pages;*.key;*.numbers are decoded for the search, the contained .xml files with the main contents (document.xml, content.xml, index.xml, ...) and in case of .pages any existing Preview.pdf are also omitted, to avoid redundant search hits.

  Files with a red X icon will not be searched, except if they are specifically targeted via a selection or tagmark.

- In NTFS, all "real" hard links (i.e. hard links other than SFN) except for one can be optionally omitted from logical searches and indexing. Nowadays on Windows installations often between 10,000 and 100,000 hard links of system files exist, for example 27 links to a file like "Ph3xIB64MV.dll" in directories such as
  \Windows\System32\DriverStore\FileRepository\ph3xibc9.inf_amd64_neutral_ff3a566…
  \Windows\System32\DriverStore\FileRepository\ph3xibc2.inf_amd64_neutral_7621f5…
  \Windows\System32\DriverStore\FileRepository\ph3xibc5.inf_amd64_neutral_22703…
  \Windows\winsxs\amd64_ph3xibc9.inf_31bf3856ad364e35_6.1.7600.16385_none_a…
  \Windows\winsxs\amd64_ph3xibc5.inf_31bf3856ad364e35_6.1.7600.16385_none_9…
  \Windows\winsxs\amd64_ph3xibc12.inf_31bf3856ad364e35_6.1.7600.16385_none_6…
  etc.
  By searching only in one hard link of a file, you can typically exclude several GB of

duplicate data and yet don't miss anything if you search all other files. Those additional hard links that are omitted are those whose hard link count is grayed out. Search hits in the only hard link that does get searched are marked with the hint "→Links" in the Descr. column to remind you of the other hard links of the same file in case those search hits are relevant.

- Option to apply logical simultaneous searches to various metadata of files in addition to the file contents. More precisely, they can be applied to the cells of any selected directory browser column such as Name, Author, Sender, Recipients or Metadata. That can spare you from pasting your keywords in the filter dialogs of various directory browser columns. That methodology is also more thorough because all the text addressed by this feature is searchable in UTF-16, whereas elsewhere the same data may be fragmented (e.g. filenames in particular in FAT), specially encoded (e.g. sender and recipients as quoted printable in e-mails), compressed, or stored in unexpected code pages. It is also convenient because any hits will be presented and listed in the same fashion as ordinary search hits in file contents, just specially marked in the search hit description column with the name of the column that the text that contains the search hits actually belongs to and highlighted in a different color. You can also filter for search hits in metadata.

  When selecting a search hit in metadata, it is automatically searched for and highlighted in Details mode, just as ordinary search hits in file contents are automatically searched for and highlighted in Preview mode.

  Note that the simultaneous search in metadata does not search in additional cell text that is displayed in a different color, such as alternative filenames and file counts in the Name column.

- Some blind spots that logical searches have in old-fashioned computer forensics software products in the several thousand dollar price range do not exist in X-Ways Forensics, as such areas on a partition can be addressed specifically, namely any transition from file slack to directly following free space, and in NTFS and exFAT also from known uninitialized (but physically allocated) tails of files to directly following free space.

Should this operation freeze on a certain file, remember the internal ID and the name of the currently processed file are displayed in the small progress indicator window. If this operation is applied to an evidence object and it crashes, X-Ways Forensics will tell you which file when you restart the program and associate it with a report table (depends on the Security Options). All that happens so that you can exclude and omit the file when trying again.

5.11 Search Hit Lists

Available only with a forensic license, when working with a case, for evidence objects with a volume snapshot. (Otherwise the Position Manager will list search hits.)

The directory browser can show search hits. To get into this display mode (search hit list instead of ordinary directory browser), click the button with the binoculars and the four horizontal lines.

It is only available for evidence objects. In that mode of operation there are four additional columns: physical/absolute offsets of the search hits, logical/relative offsets, descriptions that include the code pages in which search hits were found and hints if found in file slack, and the search hits themselves (usually with a context preview, sortable by search term, context preview not accurate for Arabic and Hebrew text or hits in UTF-8). The directory browser's grouping options have no effect when search hits are sorted by one of these three columns. The search hit description column comes with a filter that allows to focus on notable hits, user search hits, hits in a certain code page, hits in the text extraction of documents, and hits in slack space or uninitialized tail areas of files. Search hits in all variants of UTF-16 that are not aligned at even offsets are marked in the Descr. column as "unaligned", as a small hint and explanation why you can read the text only in the alignment-aware context preview of the Search hits column, and not in the text column.

Almost all commands in the directory browser context menu are available for search hit lists as well, notably the ability to copy, view, tag and comment files. The dynamic filter based on the usual directory browser columns can be used in conjunction with search hit lists e.g. to view hits in all .doc and .xls files with certain last modification dates only.

The search hit list is based on the position and level in the directory tree where you click, so that you can e.g. see all search hits in files in \Documents and Settings and subdirectories of the same, and even search hits from all evidence objects of the entire case at the same time, using the case root window. Also it's possible to conveniently select one or several search terms for search hit viewing, in the search *term* list in the Case Data window. Like that it's also an easy task to find out how many search hits there are for any given search term for any level in the case tree, as that number is displayed in the directory browser's caption based on the current search hit list.

Search hit lists are "dynamic" in that they are composed "on the fly" depending on selected search terms, explored path, current filter settings and based on the settings of the search term list (logical AND combinations and the "1 hit per item" option).

Search hits can be marked as notable (such that a yellow flag is displayed on the left) with the directory browser context menu or by pressing the Space key. With the Space key you may also remove that mark. You may unmark multiple selected search hits as notable by holding the Shift key when invoking the "Mark as notable" context menu command. You can filter for notable search hits via the “Search hits” column filter.

Search hits are stored in the metadata subdirectory of the respective evidence object. When you no longer need certain search hits, select them and press the Del key. When you no longer need any search hits of certain search terms, select the search terms in the search term list and press the Del key.

5.12 Search Term List

Displayed in the Case Data window when in search hit viewing mode (after clicking the button with the binoculars and the four horizontal lines). The search term list contains all the search terms ever search for in the case unless deleted by the user. The search terms can optionally be

sorted alphabetically in ascending order or by the listed search hit count in descending order, via the context menu of the search term list, to make it easier to locate a certain search term in lengthy lists.

Selecting search terms in the search term list and then clicking the Enter button allows you to list all the search hits for these search terms in the currently selected path, subject to filters, in the search hit list. You can select multiple search terms by holding the Shift or Ctrl key while clicking them. You may press the Del key to delete selected search terms and all their search hits permanently.

To reduce a search hit list to a list of unique files that contain at least one search hit, check "List 1 hit per item only" and then click Enter. This can be very useful if you are going to review all such files manually, ensuring that each such file is listed only once. No assumption must be made that somehow "the most useful" search hit in each file is the one that makes it to the list, or if multiple search terms are selected the one listed search hit is for a search term that you consider more important. The reduction is non-destructive. Bringing back the original, complete search hit list merely requires that you uncheck this special box and click the Enter button again.

The option to list 1 search hit per item only does not filter out search hits in slack space. This is useful because the slack of a file is typically not related to the contents of that file, so any search hits in the slack would likely have a totally different context than search hits in the logical portion of the file and thus need to be reviewed additionally. Please note that it is still necessary to unselect the "1 hit per item" option to separately check out search hits in conglomerates such as pagefile.sys and the virtual "Free space" file, which contain data from totally different sources. The "1 hit per item" option is most useful for documents, for which you can often tell after one quick look in Preview mode whether the entire file is relevant or not.

It is possible to see (and via the Export list command in the context menu copy) the hit counts for selected search terms in the search term list. These hit counts are based on the current settings for the search hit list that is on the screen, take all filters into account, the explored path, any active AND combination etc. It is the numbers of hits that are actually listed, not the numbers of hits that have been recorded/saved. To see the total numbers of hits, deactivate any filter and select all search terms. Note that the "List 1 hit per item only" option also functions like a filter for search hits.

Question: Why when all the search terms are selected with "List 1 hit per item only" are the counts returned different from when I click on each search term individually with the same setting? Answer: Because the option is "List 1 hit per item only", and not "List 1 hit per search term per item only". Many users do not understand that. Imagine if in the same file there is 1 hit for search term A and 1 hit for search term B, and you select both A and B with that option enabled, then only 1 hit is listed, either the one for A or the one for B (up to X-Ways Forensics to decide). So the displayed hit count is 1 for one search term and 0 for the other one. If then you select the other search term only and click "Enter", the count for that search term will change from 0 to 1 because that is now the only possible search term from which hits can be listed, and up to 1 search hit is listed per file, so that 1 hit is listed.

There are two ways how to logically combine multiple search terms with Boolean operators:

1) By default, multiple selected search terms are combined with a logical OR. To force a search term, select it and press the “+” key. To exclude a search term, select it and press the “-“ key. To return a search term to normal OR combination, press the Esc key. You may also use the context menu of the search term list for all that. The below examples describe the effect of selecting the search terms A and B depending on their “+” or “-“ status.

A
B
= search hits for A and search hits for B that occur in any files (normal OR combination)

+A
B
= search hits for A and search hits for B that occur in files that contain A

+A
+B
= search hits for A and search hits for B that occur in files that contain both A and B (AND)

A
-B
= search hits for A that occur in files that do not contain B

2) For a logical AND combination, if the search terms are *not* marked with “+” or “-“, you may also use the small scrollbar that appears when you select multiple search terms. Allows you to see only search hits in files that contain all the selected search terms *at the same time*. You can combine up to 7 search terms that way. If you select more than 2 search terms, you also have the option to be less strict and only specify a *minimum* number of different search terms in the same file, e.g. require that of search terms A, B, C and D any combination of two of them in the same file is sufficient, e.g. A and B, or A and C, or B and D, etc. (fuzzy/flexible AND combination).

In addition to the "Min. *x*" option, the search term list also offers offers a "Max. 1" option when multiple search terms are selected that are not forced with a + or excluded with a -. "Max. 1" will list search hits only if they are contained in files that do not contain any of the other selected search terms. For example for 3 search terms, to get the same results otherwise, you would have had to list search hits for search term A while excluding B and C, then list search hits for B while excluding A and C, and then list search hits for C while excluding A and B, which of course is not as elegant and does not show you all such singular search hits at the same time.

When 2 search terms are selected in the search term list and combined with a logical AND (using either of the two available methods), additionally you can now require that search hits must be "NEAR" to each other to be listed, to find more likely relevant combinations of both search terms in the same file, exactly like with a proximity search. The maximum distance between the search hits that constitutes "NEAR" can be defined by the user in bytes. A NEAR combination may also be applied for more than 2 selected search terms. The effect is that a search hit is listed only if *any* of the other selected search terms occurs nearby.

This paragraph quoted from wikipedia.org: The basic, linguistic, assumption is that the proximity of the words in a document implies a relationship between the words. Given that authors of

documents try to formulate sentences which contain a single idea, or cluster related ideas within neighboring sentences or organized into paragraphs, there is an inherent, relatively high, probability within the document structure that words used together are related. Where as, when two words are on the opposite ends of a book, the probability there is a relationship between the words is relatively weak. By limiting search results to only include matches where the words are within the specified maximum proximity, or distance, the search results are assumed to be of higher relevance than the matches where the words are scattered.

What's more, the search term list offers a "NOT NEAR" option (abbreviated NTNR) in addition to "NEAR". With 2 selected search terms, NTNR will ensure that only search hits are listed that are *not* located in vicinity of any search hits of the respective other search term. With more than 2 selected search terms, the results are currently undefined.

5.13 Event Lists

Available only with a forensic license, when working with a case, for evidence objects with a volume snapshot.

When extracting metadata (part of volume snapshot refinements), X-Ways Forensics can compile a list of events from timestamps that can be found at the file system level as well as internally in files and in main memory. Conceivable sources are browser histories, Windows event logs, Windows registry hives, e-mails, etc.. An event list works exactly like a search hit list and can be displayed by clicking a button which is located next to the search hit list button, with a clock icon on it. Just like a search hit list, an event list comes with additional columns: the event timestamp, event type, event category, and some events have an individual description/additional text, for example events recorded in the Windows registry and in Internet Explorer index.dat files.

If an event list is sorted chronologically, by timestamps, it works like a timeline, which may allow you to figure out a sequence of events of different kinds stored in different places (e.g. e-mail received, attachment saved, application started, document printed, file deleted) that otherwise could not be seen together in context. You may see events from different evidence objects at the same time from the case root window, explore recursively or by path, sort by event type or event category, see all the usual file properties, view files, navigate to the definition of an event within a file (if a relative offset is available) and filter for certain date ranges.

You may mark events as notable just like search hits and filter for notable events via the Timestamp column.

Event-based analysis instead of file-based analysis is a progressive new approach with a totally different perspective that may lead to knowledge about activities recorded on computers that otherwise could hardly be gained. You may see connections (related activity) that otherwise could be overlooked, and may be able to better explain the logic behind what has happened.

The sources of events that are exploited by the metadata extraction in this version include all the supported file systems (i.e. all the timestamps listed in the timestamp columns of the directory browser; modification, record update and last access are omitted if identical to the corresponding

creation timestamp), processes in supported memory dumps, extracted or processed e-mail, as well as files of these types:
index.dat
Internet browser SQLite databases
.firefox (~55) fragments
_CACHE_001_ and _CACHE_002_
.lnk shortcuts
.automaticDestination-ms
.chrome Chromium cache data_1, data_2
.usnjrnl fragments
Registry hives*
Windows .evt event logs
Windows .evtx event logs (Most extracted events come with a description that includes the event source, the event ID and the record number. The record number allows you to quickly search for the record in the HTML preview if you need further details about that particular event.)
DataStore.edb (MS Windows operating system update events)
.hbin Registry hive fragments
.doc (last printed)
.msg
rp.log XP restore point
INFO2 XP recycle bin
.recycler Vista recyle bin
.snapprop Vista volume shadow copy properties
.cookie
.gthr;.gthr2 Gatherer and Gatherer fragments
.pf prefetch
attach timestamps from EDB
signing date from EXE/DLL/SYS/...
boot time from ETL (event trace log) files
OLE2 last modification
last saved in Office documents and RTF
Skype main.db (chats, calls, file transfers, account creation, ... - you can read entire chats if sorted chronologically)
Skype Chat Sync
internal creation from miscellaneous file types, including Exif timestamps from photos
JPEG GPS
Unix/Linux/Macintosh system logs (These events are practically of significance especially for USB device history examinations.)

* More specialized events than just standard registry timestamps are output optionally when you create a registry report, depending on the report definitions used!

The event type is displayed in gray if the timestamp is a previously valid timestamp, for example such as those found in NTFS in 0x30 attributes or index records of INDX buffer slack or in $LogFile.

Timestamps from 0x30 attributes in NTFS file systems are output as events only if actually different from their 0x10 counterparts and not identical to the 0x30 creation timestamp. They are

marked as "0x30" in the Event Type column. Malware might give itself harmless looking timestamps after deployment, so that it does not seem to be related to the time of intrusion/infection. The 0x30 attribute timestamps, however, remain unaltered (except if the file is renamed or moved later), and that is the reason why some examiners are interested in them. If the time frame of intrusion/infection is known, related files would be found in the event list thanks to the original 0x30 attribute timestamps.

0x30 timestamps are marked in the event list with an asterisk if they are later than the corresponding 0x10 timestamps, which seems unnatural and in some rare cases might be the result of backdating by the rightful users of the computers themselves. Under certain circumstances, backdating documents is seen as fraudulent and illegal. However, much more commonly 0x10 timestamps predating 0x30 timestamps is just the work of installation programs or the result of copying a file or moving a file from one volume to another or extracting a file from a zip archive, where Windows or other programs artificially apply the original creation time of the source file to the destination once copying turns out to be successful (internal programmatic backdating).

The selections in the event type filter are not remembered by the program from one session to the next.

Please see the description of the timestamp columns for more information.

5.14 File Type Categories.txt

This customizable file defines of which file types categories are comprised. The name of a category is preceded by three asterisks and a space (***). Following is a list of file types that belong to that category, one per line. Such lines must start with either a “+” or a “-“, where “+” simply means that type is checked in the file type filter. After that, typical extension for that file type follows, plus a space character, followed by a description of the file type. Only lower-case letters are to be used in extensions. The same file extension/type may occur in multiple categories (see Category column description for limitations).

Alternatively to extensions, entire filenames are supported as well. This is useful for certain files with a well-defined name whose extension alone is not specific enough or which do not have any extension. Complete filenames have to be enclosed in semicolons. Examples:
-;index.dat; Internet Explorer history/cache
-;history.dat; Mozilla/Firefox browser history
-;passwd; Existing users

There is a virtual “Other/Unknown type” category, which is not specifically defined in the file and simply covers all files that do not belong to any other, defined category.

You may store additional custom definitions of file types and categories in a separate file named "File Type Categories User.txt". This file will be read and maintained in addition to the standard definitions in "File Type Categories.txt" and has the same structure, but is not overwritten by updates of the software if contained in the installation directory, so that you can easily continue

to use it even when overwriting your installation with a new version.

File types are **rank**ed by importance/relevance and you may filter by this rank. For example, filtering out those file types ranked #0 will exclude font files, cursors, icons, themes, skins, clip arts, etc. Files with a low rank are of importance just in very specific investigations, for example source code, in which you would not be interested when looking for office documents or pictures for example, but definitely when hunting a virus programmer. Higher ranked file types are relevant in more cases. Generally the rank is useful in simple cases where you can expect to find what you are looking for in file types that are fairly well known. As another idea, you could make it a habit to only index files with higher ranks.

You also have the option to assign file types to a so-called **group**, a concept that is not identical to a file type category. Useful for example if your standard procedure is to let examiner A check out pictures and videos, examiner B documents, e-mail, and other Internet activity, and examiner C operating system files of various kinds, because of their specializations. You can give these groups meaningful names and filter for them, also using the Type Status dialog window. The groups are displayed in the Type filter.

All the definitions about file type ranks and file type groups are made in the "File Type Categories.txt" file. Suggestions for ranks and an example of a group of files that may deserve special attention are already predefined. Both ranks (from 0 to 9, where missing means 0) and groups (letters from A to Z) can be optionally specified following a tab at the end of a line, in any order, for example as "2P" or "DI3". So up to 10 rank levels are possible, but it is not necessary to fully utilize this range. Up to 26 groups are possible. You do not have to start alphabetically. The case of the letters is ignored. You may also define ranks and groups for an entire category, following a tab in a category line. File types that have no rank and category inherit both from the category to which they belong.

To give a group a more descriptive name than just a single letter, insert group definition lines at the end of the text file that start with a equal sign, e.g.
=P=Photos and videos for image group
=D=Docs, e-mails and Internet
=I=File types to index

You may store additional custom definitions of file types and categories in a separate file named "File Type Categories User.txt", which will be read and maintained in addition to the standard definitions in "File Type Categories.txt" and has the same structure and is not overwritten by updates of the software if contained in the installation directory, so that you can easily continue to use it even when overwriting your installation with a new version.

5.15 Hash Database

Functionality only available with a forensic license. An internal hash database, once created, consists of 257 binary files with the extension .xhd (X-Ways Hash Database). The storage folder is selected in the General Options dialog. Such an hash database is organized in a very efficient way, which maximizes performance when matching hash values. It is up to the user to decide on

what hash type the database will be based (MD5, SHA-1, SHA-256, ...), and it is up to the user to fill the hash database with hash sets and hash values, either by creating hash sets in X-Ways Forensics yourself or by importing hash sets from other sources. The same hash database can be shared and used simultaneously by multiple users/instances if the same storage folder is selected. However, it cannot be *updated* while other users/instances are using it.

It is possible to maintain two separate hash databases at the same time, databases based on the same hash type or different hash types. Useful for example if you receive hash sets from different sources with different hash types (e.g. some with MD5 and some with SHA-1 values) and wish to use them simultaneously. The second hash database may be stored on a different drive. Useful if for example the primary hash database for general use is shared with colleagues on a network drive and the user wishes to create or import new hash sets, either for temporary use only or while the primary hash database is locked by other users, into a locally stored second database.

Each hash value in the hash database belongs to one or more hash sets. Each hash set belongs to either the category “irrelevant” / “known good” / “harmless” or “notable” / “known bad” / “malicious” / “relevant”.

Hash values of files can be computed and matched against the hash database when refining the volume snapshot. The directory browser's optional columns “Hash Set” and "Category” will then reveal for each file to which hash sets and category it belongs, if any (which allows you to sort/filter by these aspects and ignore irrelevant files easily or focus on files you are looking for). If the hash value of a file is contained in multiple selected hash sets, the program will report all matching hash sets and indicate the category of one of the hash sets. It also checks whether the matching hash sets all belong to the same category, and if not, will show a warning.

An optional second, separate hash database of *block* hash values (instead of normal file hash values), stored in a separate directory, allows you to search for incomplete remnants of known highly relevant files block-wise on other media.

Via the Tools menu you get invoke the dialog window to manage the active hash database(s), which allows you to

- start a fresh, blank hash database (and discard the existing current database, using the "Initialize" command, where you have the opportunity to select a new hash type),
- view a list of the hash sets that are contained in the database,
- rename hash sets,
- merge hash sets (note that duplicate hash values in the resulting hash set are not removed immediately, but next time when you add a hash set, and note that you are not warned if you are merging hash sets of different categories),
- toggle the category of hash sets,
- verify the integrity of the hash database,
- import selected hash set text files*,
- import all the hash set text files in a certain folder and all its subfolders (ditto), optionally into a single internal hash set whose name you have to specify,
- export selected hash sets (for example if you wish to exchange individual hash sets with other examiners, not the whole database),
- and switch between the normal file hash database and the block hash database.

* NSRL RDS 2.x, HashKeeper, and ILook text files are supported, plus hash sets in the JSON/ODATA format layout as used by Project Vic (versions 1.0, 1.1 and 1.2) as found in the Hubstream Inbox. Another import and the export format is a very simple and universal hash sets text file, where the first line is simply the hash type (e.g. "MD5") and all the following lines are simply the hash values as ASCII hex or (for SHA-1) in Base32 notation, one per line. Line break is 0x0D 0x0A.

The Create Hash Set command in the directory browser's context menu allows you to create your own hash sets in anyn of the internal hash databases. Whenever importing/creating hash sets, duplicate hash values within the same hash set will be eliminated. When importing the NSRL RDS hash database, X-Ways Forensics checks for records with the flags "s" (special) and "m" (malicious) so that these hash values are not erroneously included in the same internal hash set that should be categorized as irrelevant. The hash database supports up to 65,535 hash sets.

There is a way to efficiently delete individual hash values from an existing hash set, by importing a hash set file (simple 1-column format, 1 hash value per line), where the hash values to delete must be listed first and must be prepended with a minus sign ("-"). The file must have the same name as the existing hash set in the database that you wish to update (additional filename extension allowed).

5.16 PhotoDNA

X-Ways Forensics can employ the PhotoDNA hashing algorithm for photos, until further notice. Because of the robustness of the hash algorithm and its specialization in photos, it usually allows to automatically recognize known photos even if they have experienced lossy compression repeatedly (e.g. JPEG), if they have been stored in a different file format, resized, partially blurred/pixelated, color-adjusted or contrast-adjusted etc. Unlike hash values computed by conventional general purpose algorithms, PhotoDNA hashes are resistant to various such image alterations. Optionally, known photos can be recognized even if they were mirrored (flipped horizontally).

For licensing reasons the PhotoDNA functionality is made available as a separate download, and provided by X-Ways itself **only to law enforcement agencies**, which may use it to prevent the spread of child sexual abuse content and for investigations targeted to stop its distribution and possession. For details about PhotoDNA please see this high level technical explanation and this press information.

If the PhotoDNA functionality is present, a 4th (!) database, with PhotoDNA hash values of photos can be created and maintained within X-Ways Forensics, and photos may be matched against that hash database in X-Ways Forensics and X-Ways Investigator to identify known incriminating content.

Law enforcement agencies may want to create and share their own collections of such hash values, or import an extensive existing collection from Project Vic (JSON/ODATA format layout version 1.0, from v18.1 of X-Ways Forensics also version 1.1, from v18.2 of X-Ways Forensics

also version 1.2). You can also import PhotoDNA hash databases of other X-Ways users, you may delete hash categories that you don't need any more, and you may merge or rename categories in your database. When importing someone else's hash database, their categories of the same name will be merged with yours. X-Ways Forensics will attempt to deduplicate hash values of similar photos when adding hash values to the database. PhotoDNA hash values may also be imported if they are stored in text files, with "PhotoDNA" in the first line, followed by 1 hash value per line in hex ASCII or Base64.

Hash values can be added to the PhotoDNA hash database for pictures in the volume snapshot of an evidence object in the same way as conventional hash sets are added to a conventional hash database, using the “Create Hash Set” command in the directory browser context menu. The database is one of now four databases that can be managed with the Tools | Hash Database command. The PhotoDNA hash database is stored in a directory next to hash database #1.

Matching is part of the "picture analysis and processing" operation in Specialist | Refine Volume Snapshot.

5.17 Time Zone Concept

The following applies to WinHex and X-Ways Forensics when operated with a specialist or forensic license.

X-Ways Forensics employs its own, not Windows' logic to convert UTC timestamps to a freely chosen time zone for display in the directory browser, in report tables and exported lists. It displays timestamps independently of the time zone selected in the examiner's system's Control Panel. The display of timestamps in X-Ways Forensics may differ from Windows because in Windows a timestamp in daylight saving time is not displayed based on daylight saving time if daylight saving time is not active when looking at that timestamp.

When working with a case, the time zone selected for that case applies globally to the entire program (selectable in the Case Properties), otherwise the one selected in the General Options dialog. When working with a case, optionally it is possible to specify different time zones per evidence object, so that you can always see local filetimes even for media that were used in different time zones, if preferable. Note that the timestamps are converted for *display* only. That means, in a recursive view in the case root that covers multiple media, *sorting* is based on absolute UTC timestamps. Optionally, the actually used conversion bias can be displayed as well (see directory browser options).

Timestamps on FAT volumes are never converted as they are not available in UTC, but based on one or several unknown local time zones. Timestamps in file systems that store the time zone explicitly are converted to UTC internally and then for display purposes from UTC to a local time zone.

The time zone definitions can be adjusted, if necessary. Please note that changing these definitions in any dialog window affects the definition of time zones throughout the program.

The standard Windows conversion technique, which depends on the time zone selected in the user's system's Control Panel, is still employed…

- in File | Properties, where the timestamps of files on the user's own system can be accessed/changed,
- for the case logging feature,
- generally when operated without a specialist or forensic license, and
- when operated without the file "timezone.dat".

You can tell that either of the latter two is true if the "Display time zone" button in the General Options dialog is grayed out or not visible.

5.18 Evidence File Containers

Only available with a forensic license. The Specialist menu allows to create a new file container, open an existing one, and close the active file container. The directory browser context menu allows to fill it with selected files.

When you need to pass on a collection of selected files (even from different evidence objects) that are of particular relevance to a case, to other persons involved in that case, e.g. specialized investigators, who do not need to or must not see irrelevant files, evidence file containers may come in handy. Most file-system level metadata (name, path, size, attributes/file mode, timestamps, deletion status, classification as alternate data stream or virtual file or e-mail message or attachment, …) and especially the contents of the file are fully retained in an evidence file container. Also when a conventional (physical, sector-wise) image is overkill because you need to acquire only selected files and not entire media, containers are recommended. Evidence file containers use a special file system (XWFS) that can accomodate most metadata from conventional file systems of the Windows, Linux, and Apple world.

Evidence file containers can be interpreted, added to a case and conveniently examined like other image files, and in particular also in X-Ways Investigator [CTR], the simplified version of X-Ways Forensics for investigators that are not computer forensic examiners, but specialized in other areas such as corruption, accounting, child pornography, building laws, ... The recipient of the container can add the container to his or her own case, view the files that it contains just like in a disk partition or a conventional image, can run keyword searches, comment on files, add files to report tables, create a report, etc. Report table associations can even be exported and imported back into the original case, via case tree context menu commands. This allows to split up the workload in large cases across multiple investigators who work simultaneously and to reconcile their results.

Evidence file containers of the current format can be understood by certain computer forensic tools other than from X-Ways. Older versions of WinHex (with a specialist license or higher), X-Ways Forensics and X-Ways Investigator can also understand them. They can all read the contents of all files and show the most essential metadata (e.g. filename, path, many attributes, most timestamps, existing or deleted). To see the maximum amount of metadata, however, please use WinHex/XWF/XWI 16.3 and later.

Artificial directories can be optionally created in containers to accommodate child objects of files, for compatibility with tools that do not accept files as child objects of other files (non X-Ways tools and WinHex/XWF/XWI 15.9 and earlier). WinHex/XWF/XWI 16.0 and later (latest release, respectively) do not need such artificial directories.

When creating a container, you chose between a direct method and an indirect method to fill it. Indirect means via your own hard disk, i.e the contents of files are not copied directly into the container, but to your folder for temporary files first (cf. General Options), and only then from there into the container. This can be beneficial because it allows a resident antivirus software to intercept these files (check them for viruses, disinfect/disarm them, rename them, move/delete/lock them, etc.), so that it prevents viruses from making it into a container. The resulting container is free of known viruses (depending on the antivirus software in use) and can reasonably be passed on to and used in an environment with higher sensitivity, higher security requirements, and/or less sophisticated virus protection.

Containers can hold around 1 billion files. In order to retain in the container the source of files that originate from different evidence objects, the names of these evidence obejcts can be included in the container as the top directory level.

An optional internal designation can be specified (up to 31 characters), which will become the volume label of the XWFS file system. An optional description can also be specified (up to 60,000 characters), which will be imported as the evidence object comments once the container is added to a case in X-Ways Forensics. The description stored in the container can still be added or edited later.

Files selected in the directory browser can be added to the container that is open in the background with the directory browser's context menu. Either you copy the logical contents of a file, the logical contents and the file slack separately, just the slack, only the block selected in File mode, or merely the file system level metadata of the file. You may also specify whether child objects of selected files should be copied to the container as well, even if they are not selected themselves, either child objects of any kind of child objects (if fully checked) or only e-mail attachments (if half checked).

Optionally containers can include the data/contents of directories themselves, i.e. depending on the file system, directory entries, INDX buffers, etc. Useful if the recipient of the container is technically versed and might be interested in timestamps or other metadata in these data structures. If you choose to include directory data in a container when creating it, this has a direct effect only on directories that are selected themselves. It has an effect on the respective *parent* directory of selected items only if you enable an additional option (“Include data structures/contents of direct parent items”). This additional decision is needed because otherwise the directory data might unintentionally reveal the names and other metadata of files that were intentionally omitted from the container, e.g. for reasons of confidentiality.

If in the container you have X-Ways Forensics recreate the original path of files that are child objects of other files, then those parent files will be included in the container at least as nominally, without data, so that the child object appears with the correct path and it is clear where it comes from, just by looking at the container. Examples for such parent files are the e-mail message that a selected attachment belongs to, the zip archive that contains a selected file,

and the document that a selected picture is embedded in. With the option “Include data structures/contents of direct parent items”, the data of such files is also included in the container, even if these files were not selected for copying themselves.

Any file that is part of a volume snapshot (e.g. even individual e-mail messages if extracted) can be added to a container. Once added, a file cannot be physically removed any more, however, its exclusion can be made permanent in the container. You have the option to automatically create report table associations for files that have been added to an evidence file container.

Optionally, hash values can be stored for the files that are copied into a container. This allows to verify the integrity of the files later, after having added the container to a case, by refining the volume snapshot. The hash values are computed directly for the data as read from the original source medium (unless you copy metadata to the container only) or taken from the volume snapshot, if available.

Optionally, the preparer of an evidence file container can pass on report table associations (either all or not those created by X-Ways Forensics internally) or comments about included files with the container. Useful to not only forward a collection of files to other investigators, but also case-specific information and preliminary findings. E.g. computer specialists could add the name of the owner of a file for non-IT examiners to see, or the reason why a file was selected for inclusion in the container.

Abort operation upon read error: This option allows to abort copying files into an evidence file container upon a read error and to not include affected files partially. Useful when acquiring files from a network location and the connection might be interrupted, if you assume that if that happens you will get the connection back and will be more successful when you try again, to avoid having incomplete files in the container, which cannot be replaced with a complete copy retroactively. Available only when not filling containers indirectly.

When closing a container that is open in the background, the user is offered to compress, encrypt, and/or split it. Splitting is useful if the container is complete and relatively huge, and e.g. should be sent to someone else on CDs or DVDs. You may also find it useful to have a verifiable overall hash value for all the data in the container, which can be computed at that occasion and embedded in the target container. You can also freeze the file system in the target container that you create in .e01 evidence file format, so that it cannot be filled further even if it is converted back later to its plain state again (to a raw image).

5.19 Related Items

Only available with a forensic license.

Files/directories that have a corresponding "related" file or directory in the volume snapshot are marked in the directory browser with a small blue arrow pointing downwards on the left-hand side of their icon. A secondary tooltip appears for files with a "related" file when hovering the mouse cursor over the icon, which conveniently tells you the path and name of that related file, for example the target of a symbolic link. There are four different kinds of related objects:

1) When taking a volume snapshot of Unix-based file systems, symbolic links are connected to their targets in the volume snapshot as so-called related files, so that you can conveniently navigate to the target by pressing Shift+Backspace. Also one of potentially several symlinks pointing to a certain target will become the related file of the target, so that you can conveniently navigate to the symlink or quickly see in the first place that one or more symlinks exist that point to a certain target, since any file that has a "related" file in the volume snapshot is marked with a tiny blue arrow next to its icon. Also the same arrow will tell you whether the target of a symlink can actually be found in the file system. If a symlink links to other symlinks, those are not recursively linked. If resolving symlink takes to long because there are many symlinks in a volume, you may safely abort that step at any time.

2) When taking a snapshot of volumes with Windows installations, certain reparse points (a.k.a. junction points) are connected to their targets in the volume snapshot just like as symlinks in Unix-based file systems, so that you can conveniently navigate to the target by pressing Shift+Backspace. Also there will be a back-reference to one reparse point, so that you can conveniently navigate to that reparse point or quickly see in the first place that one or more reparse points exist that link to a certain directory, since any directory that has a "related" directoy in the volume snapshot is marked with a tiny blue arrow next to its icon. Forensic license only. Reparse points that do not get connected with their target directories will still show a comment that advises you of the target path as in earlier versions of X-Ways Forensics.

3) Hard links in HFS+ point to their so-called indirect node file.

4) Files found in volume shadow copies in NTFS point to their shadow copy host file. VSC host files point to their corresponding snapshot properties file.

5.20 External Analysis Interface

Via the menu command "Export Files for Analysis" in the CaseData window, you can send files (for example all files in the case that belong to a certain category) to an external program for further analysis. This external program must comply with the interface described below. Requires X-Ways Forensics or X-Ways Investigator or WinHex with a forensic license.

The analysis result can be imported back into X-Ways Forensics with the Report Table Import menu command in the Case Data window. (For example, right-click the case title where it is printed in bold.) That will associate files classified by the external software with certain report tables (and may create new report tables), which allows you to filter for such files or create a report about them.

For example, the software DoublePics can recognize known pictures (even if stored in a different format or altered) and return a classification such as “CP”, “relevant”, or “irrelevant”.

Technical description of the interface

All files or files in a certain category or all tagged files or all non-excluded files are copied into a

subfolder of the output folder specified by you. The subfolder is named with a CRC in hexadecimal characters that is unique for the active case. The files are named with unique IDs (64-bit integer numbers). One additional file named "Checksum" is created that contains 4 bytes with the same CRC, 4 bytes with the handle of the main window of X-Ways Forensics (or X-Ways Investigator, for that matter), 8 reserved bytes, and 128 bytes with the case title in UTF-16. When the files have been copied, X-Ways Forensics executes the external analysis program and specifies the complete path of the subfolder in quotation marks as a parameter.

The external program can now perform the analysis. It can classify files by creating one .rtd file for each classification.

When finished, the program can optionally check whether the X-Ways Forensics main window still exists and, if so, make X-Ways Forensics aware of the availability of the results, by sending a WM_SETTEXT messages to the main window, where the text starts with "Import: ", followed by the path of the directory where to find the .rtd files, without quotation marks. This will trigger the import automatically. Alternatively, the user can import the result as described above.

The names of the .rtd files (report table definition files) will be used as the report table name. An .rtd file starts with a 4-byte signature (0x52, 0x54, 0xDE, 0xF0), the 4 byte checksum (see above), followed by the 64-bit file IDs (integer numbers) that indicate the files that should be associated with that report table.

6 Volume Snapshots and their Refinement

6.1 Introduction

A volume snapshot is a database of the contents of a volume or physical medium (files, directories, …) at a given point of time. The directory tree and the directory browser present views into this database. Based on the underlying file system's data structures, it consists of one record per file or directory, and remembers practically all metadata (name, path, size, timestamps, attributes, ...), but not the *contents* of files or data of directories.

A volume snapshot usually references both existing and previously existing (e.g. deleted) files, also virtual (artificially defined) files if they are useful for a computer forensic examination (e.g. so that even unused parts of a disk or volume are covered). Operations such as logical searches, indexing, and all commands in the directory browser context menu are applied to the files and directories as they are referenced in the volume snapshot. Because of compressed files and because deleted files and the virtual "Free space" file may be associated with the same clusters of a volume multiple times, the sum of all files and directories in a volume snapshot can easily exceed the total physical size of a volume.

A volume snapshot is stored on the disk either as a set of files named Volume*.dir in the folder for temporary files or (if associated with a case) as files named “Main 1”, “Main 2”, “Main 3”, “Names”, …, in the evidence object's metadata directory.

6.2 Refinement at the Volume/Sector Level

The Specialist menu allows to *expand/refine* the standard volume snapshot in various ways, such that they contain more than referenced by the regular file system. Requires a specialist or forensic license. Full functionality only with a forensic license.

6.2.1 Run X-Tensions

X-Tensions are DLLs, which you can program yourself, to extend the functionality of X-Ways Forensics or use it automatically for your own purposes. More information.

6.2.2 Particularly thorough file system data structure search

Running a particularly thorough file system data structure search is possibly a lengthy operation, depending on the size of the volume, and for that reason not done automatically when taking the volume snapshot.

FAT12/FAT16/FAT32: Searches for orphaned subdirectories (subdirectories that are no longer referenced by any other directory).

Ext3/Ext4: Similar to the procedure for FAT. Checks the entire volume for previously existing directory structures whose contents are no longer known from corresponding inodes (these would have been looked at as part of the regular volume snapshot already). Such directories are listed with a generic name, usually in "Path unknown", but potentially in the root directory, if that is where they existed previously (the root directory is special in this situation, as it has an unchangeable ID).

ReiserFS, Reiser4: Searches for deleted files (which are not included in the standard volume snapshot at all).

UDF: While the first and the last session of multi-session UDF CDs/DVDs will be listed automatically, additional sessions in the middle can be found only with this option.

CDFS: Usually all sessions on a multi-session CD/DVDs are detected automatically. In cases where they are not (e.g. when CDFS co-exists with UDF or if the gaps between the sessions are unusually large), this will detect sessions beyond the first one.

RAM (main memory): May find terminated processes and rootkits.

NTFS: Volume shadow copies can be parsed optionally, with a forensic license. Existing and previously existing volume shadow copy host files are checked for valuable information that would not be available otherwise, such as files that cannot be found in the current $MFT any more or previous versions of files whose contents have changed. Those files will be reconstructed up to 1 GB in length according to the shadow copy. Processing of volume shadow copies, if any,

occurs before all the other operations that are part of the particularly thorough file system data structure search (parsing $LogFile, optionally searching for FILE record outside of $MFT and outside of VSC, searching for index records in the slack of INDX buffers). If there are volume shadow copies, the caption of the small progress indicator window will tell you when they are being parsed. Volume shadow copy host files that you exclude before processing will be omitted.

Files found in volume shadow copies are specially marked with "SC #" in the Attr. column, or "SC #, prev. version" if they are previous versions of files that were known to the volume snapshot already before the thorough file system data structure search, so that it is easy to filter them in or out. # stands for the sequential number of the snapshot in which these files were found. Remember you can sort by ID to see the files they are a previous version of next to them. You can also easily navigate to the VSC host by using the command Navigation | Find related file in the directory browser context menu, for example so that in Details mode learn more about that particular snapshot. You could then invoke the same command once more to navigate to the corresponding snapshot properties file, where in Details mode you learn even more, e.g. description and official creation date.

Optionally avoid that previous versions of files in volume shadow copies are added to the volume snapshot if they are exact duplicates (identical file contents) so that it is much easier to focus on files for which actually previous data is still available. Even if modification dates are different, the file contents are often the same for files installed by the operation system. If fully selected, X-Ways Forensics will compare files up to 128 MB, if half selected, only up to 16 MB, as to not waste too much time on this feature.

NTFS: FILE records can be optionally searched everywhere, in sectors that neither belong to the current MFT nor to a volume shadow copy (VSC) processed by the above-mentioned option. Such FILE records can be found e.g. in free space after a partition has been recreated, reformatted, moved, resized, or defragmented. Time consuming on very large partitions.

NTFS: With a forensic license, the current $LogFile as well as old versions of $LogFile found in processed volume shadow copies can be exploited. The contents of deleted files can often be reconstructed thanks to $LogFile. Index records remnants in $LogFile as well as in the slack of INDX buffers can be exploited that either reveal previous names or paths of renamed/moved files/directories that were known to the volume snapshot before or deleted files that the volume snapshot was not aware of before (without file contents, though). You can indicate whether you are interested in earlier names and paths of renamed/moved files and directories or not. If the checkbox for earlier names/paths is half checked, you may find earlier names/paths of renamed/moved files in the Metadata column and don't get additional files in the volume snapshot for each earlier name/path. You can also indicate whether you are interested including traces of files in the volume snapshot whose clusters are unknown and for which only name, size, timestamps and attributes are available.

During all the suboperations for NTFS, the inclusion of redundant (identical) files in the volume snapshot is avoided as much as possible. If the only new information gained from old versions of FILE records or index records is previously valid timestamps, no earlier names/paths/contents of files, or if you have indicated that you are not interested in earlier names/paths, then these timestamps are only output as events, depending on the volume snapshot refinement option "Provide by-catch timestamps from various sources as events".

NTFS: You can indicate whether you are interested in getting files included in the volume snapshot whose clusters (and therefore data) are totally unknown, with only metadata (e.g. filename, path, size, attributes, and timestamps), as may be found in index records in INDX buffers or in $LogFile. If checked, all previously existing files of which metadata only is known will be included in a volume snapshot. If not checked, those files will be ignored.

Other file systems: no action taken

6.2.3 File Header Signature Search

The "**File header signature search**" operation helps to include files in the volume snapshot that can still be found in free or used drive space based on their file header signature and are no longer referenced by file system data structures. You are asked to select certain file types for detection, specify a default file size, an optional filename prefix etc. Please see "File Recovery by Type" and the file type definitions for details. Files found with this method will be included in the volume snapshot only if there is no other file in the volume snapshot with the same start sector number yet (overwritten files don't count), to avoid duplicates. Files found with this method are listed with a generic filename and size as detected by the "File Recovery by Type" mechanism. If applied to a physical, partitioned evidence object, only unpartitioned space and partition gaps will be searched for file headers, because the partitions are treated as separate, additional evidence objects.

Usually results of the file header signature search are output in a special virtual directory for carved files, which is a subdirectory of "Path unknown". However, there is an option to show resulting files as child objects of existing files, if the carved files were found within these other files.

6.2.4 Block-wise Hashing and Matching

Available with a forensic license. Block-wise hashing may allow to identify complete or incomplete remnants of known notable files that are still floating around in free drive space even if they were fragmented and the location of the fragments is unknown, to show with some or very high certainty that these files once existed on that medium. The hash values are computed when reading from the evidence object sector-wise, and that happens at the same time when running a file header signature search if selected, to avoid unnecessary duplicated I/O, with the same sector scope. Matches are returned as a special kind of search hits. Multiple matches for contiguous blocks are more meaningful than isolated individual matches, as they are even less likely the result of some coincidence, and they are usually combined in a single hit. The size of all such hits is shown when listing search hits. The larger the size, the higher the evidentiary value of the match. Please note that X-Ways Forensics does not verify itself that contiguous matching blocks are in the same order as in the original file(s), but that can be verified manually and for data that is as unique as compressed data that is most likely the case.

Most suitable for selected notable files larger than a few sectors, files that are ideally compressed

or at least not only sparsely populated with non-zero data and do not contain otherwise trivial combinations of bytes values that occur frequently. Good examples are zip-styled Office documents, pictures and video files. Very trivial blocks within a file that consist of mostly just 1 byte value are ignored and not hashed (the same already when creating the hash set). For quicker matching, ideally work with a small hash database and do not select a hash type stronger than MD5.

Hash sets of block hashes can be created or imported in the same way as ordinary hash sets, i.e. for selected files using the directory browser context menu, but they are handled by a separate hash database for block hashes (as opposed to file hashes). That separate database is internally stored in a subdirectory of the main hash database directory. You can create hash sets consisting of the block hashes of 1 file at a time, or combined hash sets of multiple selected files. The block size is currently always 512 bytes and might be user-definable in a future version.

6.3 Refinement at the File Level

The following operations are applied *after* the aforementioned operations, to files that are already contained in the volume snapshot, and they are all applied together and file-wise (i.e. first all operations to one file, then all operations to the next file, and so on), to process files in the order of ascending internal IDs. Some of these operations may produce additional files, which will get the next higher available internal ID. Previously existing files whose first cluster is known to have been overwritten or whose first cluster is unknown are not processed except if you specifically target them via tagging. Files that are considered irrelevant based on hash matching can be automatically omitted from all further operations to save time and avoid potentially even more irrelevant files that might otherwise be extracting from them. There is also an option to omit files that are filtered out. Both options are particular powerful in that they can target even files in advance that are not yet part of the volume snapshot when the refinement starts. For example when additional files are added to the snapshot by the file header signature search, depending on the file type these files can be further processed (e.g. hashed) or not, if the Type filter is active during the later stages of the volume snapshot refinement.

6.3.1 Hash Value Computation and Matching

Hash values can be computed for files in the volume snapshot. They are not recomputed if you apply this operation again to the same files. In addition to the mere hash computation, a forensic license allows to **match** the hash values against individually selected (or simply all) hash sets in an internal hash database. The filter can then later be used to hide known irrelevant files. Files recognized as irrelevant with the help of the hash database can be optionally excluded from further volume snapshot refinement operations, which among other benefits saves time. The hash values will not be updated in the volume snapshot once computed. However, the *matching* process (looking up the hash values of files in the volume snapshot) can be repeated for the same files at any time. This will remove previous hash set matches from these files. The hash category field will be updated only, but emptied.

It is possible to compute hash values of two different hash types at the same time when refining

the volume snapshot, for general purposes or to match them against two hash databases with different hash types. If matching is selected, all hash values will be matched against any of the two hash databases whose hash type fits. That means even if the primary hash type in the volume snapshot is MD5 and the secondary is SHA-1, and hash database #1 is based on SHA-1 and #2 based on MD5, X-Ways Forensics will match the hash values accordingly. The hash types in the volume snapshot and in the hash databases do not have to be in the same order.

A forensic license allows to verify hash values that were computed at an earlier point of time, or imported from an evidence file container. The result will be output to the Messages window. Any file whose current hash value does not match the originally recorded one will be associated with a special report table for convenient review. Running the hashing volume snapshot refinement step a second time never updates the hash values that were already computed for files in the volume snapshot.

Child objects of files inherit the hash category "irrelevant" from their parents. That is possible because if an entire file is irrelevant, everything that can be extracted from that file must also be irrelevant. However, what is extracted from a "notable" file is not necessarily also notable, because perhaps only some parts or aspects of the parent file are notable. Of course, child objects of irrelevant parents will only be output if the user chooses to not omit irrelevant files from further processing in the first place.

6.3.2 File Type Verification

A forensic license allows you to **verify file types based on signatures and** various **algorithms**, i.e. detect filename/file type mismatches in all files in the volume snapshot except those whose original first cluster is known to be no longer available. For example, if someone has concealed an incriminating JPEG picture by naming it "invoice.xls" (wrong filename extension), the recognized file type "jpg" is stated in the Type column of the directory browser. For more information see the description of the columns Type and Status. The file signatures and extensions used for mismatch detection are defined in the accompanying file type definition files, which you may fully customize. It it the same database also used for file header signature searches. Please note that the link between the current data in a free cluster and a deleted file that previously was stored in that cluster and its filename is weak, so that a discrepancy between filename extension and detected type can simply be the natural result of a reallocation of this cluster to a totally different file in the meantime. If you wish to repeat the file type verification, e.g. after editing the file type signature database, be sure to check the Again option. For the status of the Type column of the directory browser, see the “Type status” column.

Most self-extracting .exe archives are internally detected by the file signature check, too. They are classified as the file type “sfx” and assigned to the category “Archives” so that they can be specifically targeted. This prevents that compressed files in such archives go totally unnoticed in an investigation. .exe archives with Zip compression can be viewed in Preview mode, other self-extracting archives need to be copied off the image and opened with an appropriate tool like WinRAR or 7-Zip.

The file signature check also reveals hybrid MS Office files, i.e. merged MS Word and MS Excel

documents that can be opened in both applications, showing different contents. A notice in the messages window will be displayed, and any detected files will be associated with a special report table. Hybrid MS Office files are a clever attempt to conceal the contents of one of the merged documents.

6.3.3 Extraction of Internal Metadata

Requires a forensic license.

a) Can check the file format consistency of EXE, ZIP, RAR, JPEG, GIF, PNG, RIFF, BMP, and PDF files. The Type Status column will show the result, either "OK" or "corrupt".

b) Allows to extract internally stored creation times from OLE2 compound files (e.g. pre-2007 MS Office documents), EDB, PDF, MS Office HTML, EML, MDI, ASF, WMV, WMA, MOV, JPEG, THM, TIFF, PNG, GZ, GHO, PGP pubring.pkr keyring, ETL, SQM, IE Cookies, CAT, CER, CTL, SHD printer spool, PF prefetch, LNK shortcut, and DocumentSummary alternate data streams. This timestamps will be shown in the Int. Creation column of the directory browser. In some cases the earliest timestamp will be extracted, which approximates the real, original creation date best.

c) Allows to copy certain file metadata to the Metadata column, which will allow you to filter by this metadata, to export the metadata with the Export List command, and to output it with a report table in a case report. Metadata can be extracted from all the file types specifically supported in Details mode plus Windows shortcut files (.lnk) and prefetch files (.pf). Only a subset of the metadata that you see in Details mode is extracted.

d) Allows to restore original file system metadata (such as filename, timestamps) when found in certain file types such as $I* recycle bin files and iPhone mobile sync backup indexes (Manifest.mbdx). Original filenames are typically much more meaningful than random names that are assigned just to guarantee uniqueness in a single directory for backup purposes. Examples of such random names are 3a1c41282f45f5f1d1f27a1d14328c0ac49ad5ae (for a file in an iPhone backup) or $RAE2PBF.jpg (Windows recycle bin). The current filename according to the file system can still be seen in square brackets in the Name column, as well as in Details mode, and the Name filter will find both the original and the current name, so that current filename is not completely lost.

Alternative names and timestamps are also extracted from Linux PNG thumbnails as known from Ubuntu and Kubuntu distributions, desktop manager MATE and GNOME ThumbnailFactory. The name of the original file is shown in square brackets in the Name column and the recorded timestamp of the original file is shown as a "Content created" timestamp. The complete path of the original file can be seen in the Metadata column.

e) Populates the Sender and Recipients columns for original .eml files.

f) Creates previews of Internet browser SQLite databases, which may require that the files have been checked for their true file type. Supports Firefox history, Firefox downloads, Firefox form

history, Firefox sign-ons, Chrome cookies, Chrome archived history, Chrome history, Chrome log-in data, Chrome web data, Safari cache, Safari feeds, and Skype's main.db database about contacts and file transfers. Creates previews also of Internet Explorer index.dat files (including artificial index.dat files compiled from individual records from various locations during the file header signature search), Internet Explorer 10's WebCacheV*.dat files, $UsnJrnl:$J, Windows Event Logs (.evt and .evtx). Also extracts browsing history information from Safari's icon database. This alternative source is very interesting because it records browsing history even when Safari is in private browsing mode. HTML previews and views of index.dat Internet Explorer browser cache/history files contain a column with the offset of the record within the file where the data of each row has been found. This offset is presented as a link. If you click it, you will automatically navigate to that offset in the corresponding index.dat file in File mode so that it is convenient to verify the information that X-Ways Forensics has extracted from the record at that location. (Note that this works correctly only if the link is not broken into 2 lines, which may happen in v8.4 of the viewer component, but not in v8.3.7. Anyway you can still navigate to that offset manually.) The HTML child objects that will be generated can not only be used internally by X-Ways Forensics for previews of the parent file. You can also view all of these tables in an external program such as your preferred browser or in MS Excel, by sending these child object to the program of your choice (directory browser context menu). You may have X-Ways Forenscis split HTML tables after an arbitrary number of rows. You can set this number much higher if you do view the HTML previews externally with your preferred Internet browser and not with the viewer component, which cannot deal with very large tables. The existence of HTML child object with searchable text for browser data, event logs and more data sources also improves effectiveness of searches and indexing.

g) Extracts tables from various other SQLite databases in TSV format and uses the first one as a preview of the SQLite database file itself.

h) Extracts the original revision of PDF documents that were edited, if available, as a child object.

i) Provides timestamps from the file system as events to analyze in an event list.

j) Provides internal timestamps in files as events.

6.3.4 Archive Exploration

A forensic license allows to include the contents of **ZIP, RAR**, ARJ, GZ, TAR, 7Zip, and BZIP **archives** in the volume snapshot, so that files in such archives can be separately listed, examined, searched, etc., in their decompressed state, as long as the archives are not encrypted. Theoretically, there is no limit to the number of nested levels that can be processed (i.e. archives within archives within archives…). If the files are encrypted in the archive, they are marked with “e” in the attribute column and the archive itself with “e!”. This allows to easily focus on such files using the attribute filter.

Document files of MS Office 2007/2010/2013, LibreOffice, OpenOffice, and iWork are typically Zip archives, too, technically, and if so are processed in the same way by default. You can choose

to not process those files if you or the recipients of evidence file containers that you prepare only wish to see the documents as a whole, no embedded pictures or XML files separately, and don't need to extract metadata from these XML files and can recognize nested documents (documents embedded in other documents) themselves if necessary. There are many, many other file types that are technically subtypes of Zip that are processed optionally. Zip subtypes whose contents are usually irrelevant are for example .jar, .apk and .ipa, though special interest groups like malware investigators might think otherwise, so the choice is yours.

Note that for Zip archives with non-ASCII characters in filenames to be processed correctly, you need to pick the correct code page in the case properties first. E.g. for Zip archives created under Linux, that's likely UTF-8. For Zip archives created under Windows with WinZip, that's likely a regional code page.
Note also that split/spanned/segmented archives are not supported.

6.3.5 E-mail Extraction

A forensic license allows to separately list and examine **e-mail messages** and e-mail **attachments** stored in the following e-mail archive file formats: Outlook Personal Storage (.pst), Offline Storage (.ost), Exchange (.edb, Exchange 2010 and earlier supported, 2010 still in a testing stage), Outlook Message (.msg), Outlook Template (.oft), Outlook Express (.dbx), Kerio Connect (store.fdb files that can be processed like ordinary PST/OST files), AOL PFC files, Mozilla mailbox (including Netscape and Thunderbird), generic mailbox (mbox, Unix mail format), MHT Web Archive (.mht). By default, X-Ways Forensics tries to extract from files matched by this filter expression: *.pst;*.ost;*.edb;*.dbx;*.pfc;*.mbox;*.eml;*.emlx;*.mht;*.olk14MsgSource;*.msg;*.oft;*.mbs;store.fdb.

E-mail messages are usually output as .eml files. To conveniently focus on all extracted e-mail messages from all e-mail archives (and even processed original .eml files) it is recommended to explore recursively and use the Attribute filter (not the Type or Category filter).

The timestamp in the "Date:" line in an e-mail message's header (if accompanied by a time zone indicator like -0700 or +0200) is listed as the creation date & time. The timestamp in the "Delivery-Date:" line (or alternatively, if not available, the first "Received:" line) is listed as the last modification date & time. For extracted e-mails and their attachments, sender and recipient will be displayed in the corresponding columns in the directory browser. You may filter by dates as well as sender and recipient.

Attachments and embedded files are extracted, too, if found in the e-mail archive (exception e.g. AOL PFC) and usually become child objects of their respective containing e-mail messages in the volume snapshot. All extracted e-mails and attachments actually reside in the evidence object's metadata subdirectory and may utilize a lot of drive space.

E-mail extraction from PST can process password-protected PST archives without the password! It supports the following code pages for encoded PST files: ISO8859-1, ISO8859-2, ISO8859-3, ISO8859-4, ISO8859-5, ISO8859-6, ISO8859-7, ISO8859-8, ISO8859-9, ISO8859-10, ISO8859-

11, ISO8859-13, ISO8859-14, ISO8859-15, ISO8859-16, koi8-r, koi8-u, 1250, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 874, UTF16, UTF32, UTF8

In certain old AOL PFC files, pictures may be embedded in e-mail messages in a special way. In that case, such an e-mail message will be marked with a paperclip icon, but the picture will not be separately extracted. The picture, if JPEG or PNG, can be found, however, when extracting JPEG and PNG files from *.pfc.

Some advantages of the .eml format for output: E-mail messages output as .eml files are represented as simple and as authentic and universal as it gets. They are easy to understand, clearly structured into header and body, and extremely easy to completely view in a variety of simple programs (e.g. text editor, word processing, Internet browser, free e-mail clients like Thunderbird and Windows Mail). No commercial software like MS Outlook needed is needed to view .eml files. .eml is the "natural" format of e-mail, just like a raw image is the natural format of a disk image, if you even want to call it a "format" (actually it has no additional format specifications, it's just a plain representation of the data that it should represent). An .eml file contains the complete original metadata of the e-mail message, fully intact, exactly as it was sent and delivered. You have complete control over the file if you copy it out for someone else, can see all data, can verify that no unintended data made it into the file. You can easily redact any text in the body manually with a simple text editor, redact any metadata in the header, easily retroactively remove any attachment using a simple text editor if needed, all of which is impossible to do with a complex proprietary binary file format such as MSG. The general format of .eml files can be understood by anyone, and it is simply a text file. The format of MSG files can be understood only with a computer science or programming background, and learning it takes a lot of time. Redacting e-mail data hidden in MSG files is difficult.

6.3.6 Uncovering Embedded Data

Forensic license only. Allows to carve files of various types that are embedded in files of other various types, through a byte-level file header signature search within certain files. This is successful if the outer file (host file) is intact and the embedded file is not stored in the host file in a fragmented manner. Otherwise the embedded files may appear as corrupt. Notably this function searches for JPEG and PNG pictures, even JPEG pictures in other JPEG files (those that contain thumbnails of themselves). The files found this way will be generically named as "Embedded 1....jpg", "Embedded 2....png", etc.

This function also extracts .emf files embedded in multi-page printouts (.spl spooler files). .spl files that contain a single .emf file only can be viewed directly with the viewer component. Also extracted this way are .lnk shortcut files from .customdestinations-ms jumplists. In general, X-Ways Forensics tries to carve files of those types that in the File Header Signatures Search.txt file are marked with the "e" flag. That means you can have X-Ways Forensics uncover many more file types in other files than it does by default if you like! It carves flagged file types in those host files specified by the file masks in an edit box for which no special internal algorithm exists.

Special internal algorithms exist that properly extract, by following the data structures in the respective file format, even if fragmented, .lnk shortcut files from .automaticdestinations-ms

jump lists, files of various types from OLE2 compound files (e.g. MS Word .doc, MS PowerPoint .ppt), Firefox browser caches (based on "_CACHE_MAP_" files), Chrome browser caches (based on "index" files), Safari browser caches, Norton Backup files (N360 backup, .nb20) and Windows Vista/7 Windows.edb databases (from the latter even e-mail messages), and pictures that are embedded as Base64 in VCF files (electronic business cards).

Also extracted are thumbnails from thumb*.db files, from Google's Picasa 3 image organizer and viewer software (thumbindex.db and related files), Photoshop thumbnail caches (Adobe Bridge Cache.bc), Canon ZoomBrowser thumbnail collections (.info), and Paint Shop Pro caches (.jbf). Thumbnails in certain very old "thumbs.db" files cannot be displayed correctly. Such thumbs.db files will be assigned to the report table "Unsupported thumbs.db" and can be viewed e.g. with the freely available program "DM Thumbs" by GreenSpot Technologies Ltd. Thumbcache*.db files of Windows Vista and later are targeted indirectly if thumbcache_idx.db is in the mask and if that file is available in the same directory. That speeds up the extraction and avoids the output of numerous duplicate thumbnails (only the highest available resolution is output). If thumbcache_idx.db is in the mask, that also means that thumbcache*.db files that are specifically selected or tagged for processing are not processed unless the thumbcache_idx.db file is also selected/tagged.

Also, from PDF documents it extracts any kinds of files that are marked as embedded plus JPEG and JPEG 2000 plus Acrobat form files in XML format plus JavaScript objects (the latter may make it easier to determine whether a PDF file should be considered malware). Extracts individual cookie files from Firefox and Chrome SQLite databases, also data blocks embedded as Base64 in XML-formatted PLists (.plist) and raw data blocks embedded in binary PLists (.bplist). It is recommended to verify file types at the same time so X-Ways Forensics can distinguish between traditional (XML-formatted) PLists and binary PLists (BPLists). Many PLists do not have a .plist extension and need to be identified as PLists first. Since the type of the embedded data is not identified by the PList as such, the output also benefits from a simultaneous file type verification. Nested PLists (PLists embedded in PLists) will also be identified and processed recursively. Another child object created for PLists represents parsed text in a human-readable way and serves as a preview of the PList itself.

Also reconstructs e-mail messages and extracts contact and account information from the Livecomm.edb database, which is used by the Windows Mail client (Windows 7 and newer), and contacts from Windows Live Mail contacts.edb database, also contacts from Windows Live Messenger's contacts.edb database..

You can also uncover various potentially relevant resources in 32-bit and 64-bit Windows PE executables (programms and libraries) as child objects, in particular RCDATA, named objects, bitmaps, icons and manifests. Useful for example for malware analysis. This does not happen automatically, only if you specifically target executable files via a suitable series of file masks.

Fully Base64-encoded files in the volume snapshot, provided that they have "b64" in the Type column can be automatically decoded, and the result is output in binary as (surprise) a child object.

Last not least this function can decompress most hiberfil.sys files and automatically add the result to the case as raw memory dumps. All other files produced by this function are added to the

volume snapshot as child objects of their respective host files in which they were found. Files smaller than 65 bytes are not touched, for performance reasons.

Two separate file masks are maintained for uncovering embedded data in various file types. The second mask is optional and labelled as "special interest". For example malware investigators may choose to also process executable files that way when needed. You may prepend any element of a mask with a colon to temporarily exclude it, but keep it in the list for future reference. E.g. :*.jpg means *not* files with jpg as the extension or type.

File header signature search in all files not processed above

A separate sub-operation optional allows you to freely carve any kind of file within any file that is not processed by the first sub-operation. This is not limited to file types that are marked with the "e" flag. Use great caution to avoid delays and copious amounts of garbage files (false positives) and duplicates. Please apply this new function very carefully and only with a good reason to specifically targeted files only, such as swap files or storage files in which backup application concatenate other files without compression, not blindly to all files or random files. Remember with great power comes great responsibility.

Signatures marked with the "E" flag (upper case) are never carved within other files, to prevent the worst effects, for example MPEG frames carved within MPEG videos, zip records carved within zip archives, .eml, .html and .mbox files carved within e-mail archives, .hbin registry fragments carved within registry hives. If you know what you are doing, of course you could remove the E flag.

There is an option to apply the carving procedure recursively, that is to also carve in files that were already carved within other files themselves. This can lead to many duplicates if the outer file at level 1 is carved too big so that files can be carved in it that were also carved at level 0 (the original file).

For situations were you want to carve embedded files that are not aligned at 512-byte boundaries in the original file, you may make use of the extensive byte-level option. Files are never carved in $MFT.

The default settings will make X-Ways Forensics conduct a file header signature searches at the byte level within pagefile.sys files, to find e-mail fragments, .lnk shortcut files, pictures, etc.

6.3.7 Extraction of Video Stills

A forensic license allows to **extract** JPEG **pictures from video** files, in a user-defined interval (e.g. every 20 seconds) that can be dynamically based on the play length of the video. This functionality is applied to files whose type matches the specified file mask series. Requires an external program, either MPlayer or Forensic Framer, and requires that the volume is associated with the active case. Pictures can be extracted from all the video formats and codecs supported by MPlayer. Useful if you have to systematically check many videos for inappropriate, illegal, or otherwise relevant content (e.g. child pornography or terrorist training camp instructions). The

use of intervals ensures that you won't miss important parts of videos that are hidden in the middle of a harmless vacation or birthday party video.

Extracting pictures considerably reduces the amount of data, and looking at stills in the gallery is much faster, efficient and more comfortable than having to watch all videos one after the other. The potentially time-consuming extraction process can be run unattended e.g. over night beforehand.

Also useful if you need to include extracted pictures in a printed report. The first extracted picture at the same time optionally can serve as a preview picture for the video file in Preview and Gallery mode. ASF/WMV videos protected with DRM cannot be processed and are consequentially marked with e! in the Attr. column. Note that you may hear occasional sound from the videos. Please turn off sound on your computer if you wish to avoid this. Note also that if you select a small interval (like smaller than 5 seconds), you may not necessarily get additional pictures. This depends on how the video was encoded/compressed. Duplicate stills are omitted when extracting pictures with MPlayer.

Once JPEG pictures have been exported from videos, the videos can optionally be dynamically represented in the gallery, with all extracted stills, showing them stills in a loop, to give a much more complete impression of the contents of videos without further user interaction (without having to explore them). Thus an alternative efficient way to review a large number of videos is this: Explore recursively, filter for videos, sort in descending order by number of child objects (so that videos with a similar number of stills are shown together), and activate Gallery mode. Watch the various video stills for each video. Proceed to the next gallery page when you are confident that no incriminating videos are represented on the current page, for example when all stills have been shown, which you will know is the case when the gallery has rotated back to the first still for each video.

6.3.8 Pictures Analysis and Processing

A forensic license additionally allows to compute the percentage of **skin colors** in pictures and to detect **black & white pictures**. This can be done for the file types JPEG, PNG, GIF, TIFF, BMP, PSD, HDR, PSP, SGI, PCX, CUT, PNM/PBM/PGM/PPM, ICO. The detection of black & white or gray-scale pictures is useful when looking for documents that were scanned and faxes that were stored electronically. A forensic examiner who has to look for traces of child pornography can sort pictures by skin color percentage in descending order to immensely accelerate the job. Checking the mass of 0%..9% skin color percentage pictures (e.g. thousands of browser cache garbage files) may not be necessary any more as the most likely incriminating files will be sorted near the top of the list. Please note that there may be false positives, i.e. skin-like colors of a non-skin surface. Pictures that cannot be correctly scanned for their color contents, e.g. because they are too large or corrupt, will be listed with a question mark instead of the skin color percentage. Pictures with very small dimensions (width or height no more than 8 pixels, or width and height no more than 16 pixels each) will be marked as irrelevant with the assumption that they cannot contain incriminating pornography or documents.

For large JPEG, PNG, GIF and TIFF files, at the same time when analyzing the colors in the

pictures during volume snapshot refinement, X-Ways Forensics can optionally also create thumbnails in advance for much quicker display updates in Gallery mode later. Internal thumbnails are only created if no original thumbnails are embedded in the files and extracted at the same time, and they are actually utilized for the gallery only if auxiliary thumbnails are enabled (see Options | General). (To discard all internal thumbnails, but keep the computed skin color percentages, you may delete the file "Secondary 1" in the "_" subdirectory of an evidence object behind X-Ways Forensics' back, i.e. when the evidence object is not currently open.

If you have an internal PhotoDNA hash database, known photos can be recognized automatically even if visually altered. If you select more strict matching (allow less variation in a picture), the process can be noticeably faster in huge databases. Any resulting matches can be seen and filtered in the combined Analysis column. Please note that photos that are recognized via PhotoDNA already are not additionally checked for the amount of skin tone.

It is possible to more conveniently match pictures against the PhotoDNA hash database again, for example after having added some hash values to the database or after having assigned hash values to different categories, thanks to a new check box simply labelled "Again". You can still uncheck the "Already done?" check box for the whole picture analysis and processing operation to also discard the results of the skin color computation and precomputed thumbnails and regenerate both plus the PhotoDNA matches from scratch.

Matching pictures against the PhotoDNA hash database another time is much faster if during a previous run you have X-Ways Forensics store the computed PhotoDNA hashes in the volume snapshot. Saves the time to read the files from the disk/image again and to decode/decompress the JPEG data or other formats again (time-consuming for high-resolution photos) and to recompute the hash values. Please note that PhotoDNA hashes require considerably more drive space than ordinary hashes. Also, more than one PhotoDNA hash may be required for just one picture. It is recommended to store the hash values in the volume snapshot for future fast re-matching only if you expect your PhotoDNA hash database to change during processing of a case, for example if it is likely that you or your colleagues discover further relevant pictures in that case, forcing you to search for other copies of these pictures.

Please note that with the "Again" option when re-using previously computed PhotoDNA hashes, changes to the state of the check box "Recognize pictures even if mirrored" have no effect. That means if previously unchecked when hash values were computed for the first and stored in the volume snapshot, checking it later when re-using the stored hash values won't do any good.

To discard stored hash values you can either take a new volume snapshot, or alternatively you may delete the file "PDNA" in the "_" subdirectory of the evidence object, where the volume snapshot is internally stored.

6.3.9 FuzZyDoc

The so-called **FuzZyDoc™** technology can help you to identify known documents (word processing documents, presentations, spreadsheets, e-mails, plain text files, ...) with a much more robust approach than conventional hash values. Even if a document was stored in a different file

format (e.g. first PPT, then PPTX, then PDF), it can still be recognized. Internal metadata changes, e.g. after a "Save as" or or after printing (which may update a "last printed" timestamp), do not prevent identification either. Very often even if text was inserted/removed/reordered/revised, a document can still be recognized. This is achieved by using fuzzy hashes.

FuzZyDoc hash values are stored in yet another hash database in X-Ways Forensics. Hash sets based on selected documents can be added to the FuzZyDoc database exactly like hash sets can be created in ordinary hash databases, and the FuzZyDoc hash database can also be managed in the same dialog window as the other hash databases. For each selected document you can create 1 separate hash set, or you can create 1 hash set for all selected documents. Up to 65,535 hash sets are supported in a FuzZyDoc hash database.

FuzZyDoc is available to all users of X-Ways Forensics and X-Ways Investigator (i.e. not only law enforcement like PhotoDNA). FuzZyDoc should work well with documents in practically all Western and Eastern European languages, many Asian languages (e.g. Chinese, Japanese, Korean, Indonesian, Malay, Tamil, Tagalog, ..., but not Thai, Divehi, Tibetan, Punjabi, ...), and Middle Eastern languages (e.g. Arabic, Hebrew, ..., but not Pashto, ...). Note that numbers in spreadsheet cells are not exploited by the algorithm, only text. Note that only files with a confirmed or newly identified type will be matched against the FuzZyDoc hash database. For that reason, file type verification is applied automatically when FuzZyDoc matching is requested.

Documents whose contents are largely identical (e.g. invoices created by the same company with the same letterhead) are considered similar by the algorithm even if important details change (billing address, price, product description), depending on the amount of identical text. That means that if you have 1 copy of an invoice of a company, matching against unknown documents will easily identify other invoices of the same company. For every document that is matched against the database, up to 4 matching hash sets are returned, and the 4 best matching hash sets are picked for that if more than 4 match. For every matching hash set, X-Ways Forensics also presents a percentage that roughly indicates to what degree the contents of the document match the hash set. Two different percentage types are available. A percentage based on the total text in the processed document gives you an idea of how much of the text in the document is known/was recognized, whereas a percentage based on the text represented by the hash set gives you an idea of how closely a document resembles the original document that the hash set is based on (makes sense only if you generate 1 hash set per document, i.e. do not combine multiple documents in 1 hash set). The matching percentage does not count characters one by one, and it works only on documents that actually make sense, not on small test files that only contain a few words.

Before matching files against the FuzZyDoc hash database (a new operation of Specialist | Refine Volume Snapshot), you can specify which types of files you would like to analyze, and you can unselect hash sets in the database that you are temporarily not interested in. Note that processing less files (e.g. by specifying less file types in the mask) of course will require less time, proportionally, but selecting less hash sets for matching as such does not save time. You may specify a certain minimum percentage that you require for matches (15% by default) to ignore insignificant minor similarities. That option is not meant to save time either.

In order to re-match all documents in the volume snapshot against the FuzZyDoc hash database, please remove the checkmark in the "Already done" box first. Otherwise the same files will not

be matched again, for performance reasons. Re-matching the same files may become necessary not only if you add additional hash sets to your FuzZyDoc database, but also if you delete hash sets, as that invalidates some internal links (if that happens, it will be shown in the cells of the result column).

Matches with the FuzZyDoc database are presented in the same column as PhotoDNA matches and skin color percentages, called "Analysis". A filter for FuzZyDoc matches is available. FuzZyDoc should prove very useful for many kinds of white collar crime cases, most obviously (but not limited to) those involving stolen intellectual property (e.g. software source code) or leakage of classified documents.

6.3.10 Detection of Encryption

A forensic license allows to optionally perform **file format specific and statistical encryption tests**. With an entropy test, each existing file larger than 255 bytes is checked whether it is fully encrypted. If the test is positive (the entropy exceeds a certain threshold), the file is flagged with "e?" in the attribute column, to indicate that it might deserve special attention. Typical example: Encrypted container files, which can be mounted by encryption programs like TrueCrypt, PGP Desktop, BestCrypt, or DriveCrypt as drive letters. The entropy test is not applied to ZIP, RAR, TAR, GZ, BZ, 7Z, ARJ, CAB, JPG, PNG, GIF, TIF, MPG, and SWF files, which are well-known to be compressed internally and therefore almost indistinguishable from random or encrypted data. This test is not needed to detect that files are encrypted at the NTFS file system level or inside archives. Secondly, documents with the extensions/types .doc (MS Word 4...2003), .xls (MS Excel 2...2003), .ppt, .pps (MS PowerPoint 97-2003), .mpp (MS Project 98-2003), .pst (MS Outlook), .docx (MS Word 2007...2010), .xlsx (MS Excel 2007...2010), .pptx, .ppsx (MS PowerPointer 2007-2010), .odt (OpenOffice2 Writer), .ods (OpenOffice2 Calc) and .pdf (Adobe Acrobat) are checked for file format specific encryption; MS Office documents also for digital rights management (DRM) protection. If positive, these files are flagged with "e!" in the attribute column. This check requires that the separate viewer component is active.
Additionally, the encryption test can detect eCryptfs-encrypted files (files stored by the Enterprise Cryptographic File System for Linux), with a test that is based on eCryptfs implementations for Ubuntu 8.10, 9.04, 9.10 and 10.04. Such files will by marked with "E" in the Attributes column, just like EFS-encrypted files in NTFS.

6.3.11 Indexing

Available only with a forensic license. Reads the data with the same logic as a logical search, with the same advantages (see that topic).

Creates indexes of all words in all or certain files in the volume snapshot, based on characters you provide, based on the Unicode character set and/or up to two code pages that you select. It is possible to have up to three such indexes per evidence object (e.g. Cyrillic characters indexed in Unicode and two Cyrillic code pages). X-Ways Forensics allows you to conveniently select characters from more than 22 languages for indexing. Currently, most European and many Asian languages are predefined, e.g. German, Spanish, French, Portuguese, Italian, Scandinavian

languages, Russian, South Slavic languages, Eastern European languages, Greek, Turkish, Hebrew, Arabic, Thai, Vietnamese. You may specify each character individually, or ranges of characters (e.g. a-zA-Z) if the edit box for the character pool if the edit box starts with "range:". To index the dash itself (not recommended), specify it as the last character.

Indexing is a potentially time-consuming process and may require a large amount of drive space (rule of thumb for default settings and average data: 5-25% of the original amount of data). However, the index will allow you to conduct further searches very quickly and spontaneously. The index files are saved in the subdirectories of the metadata folder of the corresponding evidence object. The scope of the index, i.e. which files are to be indexed, can be fine-tuned. Note that the index of partitioned media such as physical hard disks solely covers unpartitioned areas. That's because each partition can have its own index.

Words shorter than a lower limit you specify are ignored. The longer the minimum length in characters, the smaller the index and the faster the indexing procedure. The default lower limit is 4 characters. Frequent irrelevant words can be excluded from the index in the exception list with a minus prefix (e.g. -and, if 3-letter words are already accepted), which reduces the size of the index and the time needed to create it. The larger the range of accepted word lengths, the larger the index becomes and the more time indexing takes. Important 3-letter words can be added to the exclusion list with a plus prefix (e.g. +xtc), which overrides the default lower limit of 4 characters. The exception list does not have to be sorted alphabetically. Words in the exception list longer than the upper limit you specify are truncated in the index. Words in the exception list are bound by the character pool and cannot contain different characters.

X-Ways Forensics can optionally distinguish between uppercase and lowercase letters, i.e. create a case-sensitive index. This can be useful e.g. if you create the index for the purpose of later exporting a word list for a customized dictionary attack.

If you have X-Ways Forensics include substrings in the index, this will further slow down index creation (by a factor of 3 to 5) and inflate the index, however, you will later be able to find e.g. "wife" in "housewife" and "solve" in "resolve". If you do not include substrings in the index, it will still be possible to search the index for substrings later, but the result will be incomplete, and the search speed much slower. Please note that it is the responsibility of the user to enable substring indexing if the words in the language to index are not delimited with spaces (e.g. in Chinese, Japanese or Thai).

Indexing will be unnecessarily slow if the data to be indexed resides on the same disk with the case file and directory, where the index is created. Try to avoid indexing with an active Internet connection if your Windows system is configured to download updates and reboot automatically upon installation.

Optionally, text in certain file types can be decoded for indexing (cf. Logical Search), and it is possible to create indexes for multiple selected computer media/images associated with a case in a single step. You can index in up to six different code pages simultaneously.

It is possible to define a character substitution list in Unicode that causes certain letters to be indexed as other letters (e.g. "é" as just "e"). This will allow you to find certain spelling variations with a single index search, e.g. both the name "René" with an accented e at the end and

"Rene" without, with either spelling. This list must have the structure
é>e
è>e
à>a
...
(i.e. 1 substition per line) and needs to be present as a Unicode text file named "indexsub.txt" that starts with the LE Unicode indicator 0xFF 0xFE. "indexsub.txt" is an optional file and expected in the X-Ways Forensics installation directory.

You will be warned if you define a space character as part of words. That is because space characters are meant to delimit words, they are not part of the words themselves. If a space character is defined to be part of words, that means a whole sentence like "Mike Smith lost his credit card today." is considered just a single word.

You can delete all indexes for an evidence object by removing the "Already done" check mark in the Refine Volume Snapshot dialog. This will also clear the "i" flag from all indexed files in the volume snapshot.

Search in Index: After indexing files, you may search the index for keywords very quickly, using the Simultaneous Search function. Select "Search in Index" from the drop-down box at the bottom. Anything in excess of the maximum word length used for indexing is ignored (so that "ridiculous" is found in the index even if in the index that word was truncated to "ridicul" based on a maximum word length of 7 letters). X-Ways Forensics does not distinguish between uppercase and lowercase letters except if a case-sensitive index was created. In a search hit list populated by an index search, physical offsets are not available.

You may conveniently run non-GREP index searches for search terms that contain space characters, just like in conventional searches. This is very important for names (e.g. "John Doe" or "XYZ Technology Ltd") and spaced compound words (e.g. "bank account" or "credit card limit"). This works even if the individual components of the compound already exceed the maximum word length that was indexed (by default 7 characters), so that you will have no trouble finding "basketball positions" (10+9 letters) or "skyscraper architecture" (10+12 letters). Just as always the components are only matched up to the length that was indexed, which is not a big problem because there are not many words other than "basketball" and "skyscraper" that start with "basketb" or "skyscra", respectively. In fact the spaces in the search terms match unindexed word delimiters other than spaces as well, such as hyphens, so you will also find "Spider-Man" and "freeze-dried" when searching for "spider man" and "freeze dried", or underscores as in "bank_account" (think of a filename like "bank_account.html"), or plus signs as in "credit+card" (e.g. common in Google search URLs when searching for more than 1 word), or periods as in "interview.pdf". So in that respect index searches are even more powerful than conventional searches. Defining spaces as being part of words is a big no-no.

6.4 More Information about Volume Snapshot Refinement

Should processing freeze on a certain file, remember the internal ID and the name of the currently processed file are displayed in the small progress indicator window. If this operation is

applied to an evidence object and it crashes, X-Ways Forensics will tell you which file when you restart the program and associate it with a report table named “Reason for crash?” (depends on the Security Options). All that happens so that you can exclude and omit the file when trying again. It does no harm (does not create duplications and does not cost much time again) if you restart snapshot refinement for that volume from scratch, as already processed files will quickly be skipped, up to the point where the refinement progress was last saved, which depends on the auto-save interval of the case.

If the hash value for a problematic (crashing) file was computed, that file and identical files are skipped automatically if you (continue to) refine the volume snapshot and compute hash values (at least if the protection against identical crasher files is active in the properties of the case). To make the case forget previous crasher files, click the Delete button in the case properties. Skipped files are also automatically added to the aforementioned report table.

You may schedule a simultaneous search in advance for the time after the volume snapshot refinement.

6.4.1 Interdependencies

There are various interdependencies between all these operations. For example, if the contents of archives are included in the volume snapshot, among these files there could be pictures that are to be checked for skin colors, or documents that are to be checked for encryption. You can work under the premise that if an additional file is added to the volume snapshot or if the true type of a file is detected as part of Refine Volume Snapshot, all the appropriate other operations are applied to that file, *if they are all selected*. The output of one operation automatically becomes the input of all other operations (or even the same operation again), where suitable.

Imagine someone tries to conceal an incriminating JPEG picture by embedding it in a MS Word document, misnaming that .doc file to .dll, compressing that file in a Zip archive, misnaming the .zip file to .dll, compressing that .dll in another Zip archive, misnaming that .zip file again to .dll, and then sends this .dll file by e-mail as an attachment using MS Outlook. If all the respective options are selected, Refine Volume Snapshot does the following: It extracts the e-mail attachment from the PST e-mail archive. It detects that the .dll attachment is actually a Zip archive. Then it includes the contents of it in the volume snapshot, namely a file with the .dll extension. That file is found to be actually another Zip archive. Consequently that archive will be explored, and the .dll file inside will be detected as a .doc file. Searching for embedded pictures, X-Ways Forensics finds the JPEG file in the .doc file and can immediately check it for skin colors if desired. All of this happens in a *single* step.

6.4.2 Notes

Except for indexing, X-Ways Forensics conveniently remembers for each and every file in the volume snapshot which refinement operations have already been applied to it, so that the file will not unnecessarily be processed again, which would lead to undesirable duplication of child objects, waste of time etc. It does not remember the individual suboptions of each operation (e.g.

whether "Create previews of browser databases" was selected for the metadata extraction) and cannot catch up on these suboptions individually. If for any reason you wish to apply certain operations again to the same file (e.g. then with different suboptions or after having updated the signature database for file type verification), you may reset a file to the state of "still to be processed" by volume snapshot refinement, by selecting it and pressing Ctrl+Del. This will also clear any computed skin color percentages, extracted metadata, hash values, hash matches, etc. However, this function does not remove any child objects from the volume snapshot. That would have to be done by the user separately, if desired, by hiding and removing them. Neither does this function delete any events that were created during prior refinement operations.

Whether a file should be processed by volume snapshot refinement or not is decided only at the time when it is that file's turn, not when you start the operation. That means if you continue to work in the program while a volume snapshot refinement is ongoing, and alter or activate or deactivate filters or tag or untag files or exclude or include files, that may still affect the scope of the operation, depending on the chosen options and depending on whether the files that you tag/untag/exclude/include/... still have to be processed or not. So if for example you find out that the operation takes too much time, you can still make the filter more strict or untag certain very large files etc., without interrupting the process.

Certain previously valid timestamps of files are output as events during various suboperations of the particularly thorough file system data structure search on NTFS, depending on the refinement option "Provide by-catch timestamps from various sources as events", which may also effect other operations whose primary purpose is not the retrieval of timestamps/events.

7 Some Basic Concepts

7.1 Edit Modes

The info pane displays for each file/disk, in which mode it was opened in the program. The info pane's context menu allows to selectively change the edit mode of the active window.

Read-only/View mode: Recommended for computer forensic examinations. In order to enforce strict forensic procedures, the only mode available in X-Ways Forensics, except for files in the current case's directory and in the general folder for temporary files, to allow to decode, decrypt, and convert them, etc. Files or disks that are opened in view mode cannot be (intentionally or accidentally) edited/altered in WinHex, only viewed. In other words, they are opened write-protected = read-only by WinHex.

Default edit mode: Modifications to files or disks opened in default edit mode are stored in temporary files. Those temporary files are created and maintained dynamically when needed. Only when you close the edit window or use the Save menu command the File Menu, the modifications are flushed and the original file or disk is updated, after prompting the user.

In-place edit mode: Please use caution when opening files or disks in in-place edit mode. All

kinds of modifications (keyboard input, filling/removing the block, writing clipboard data, replacements, ...) are written *to the original file or disk* ("in-place") *without prompting*! It is not necessary to save the file manually after having modified it. Instead, the modifications are saved lazily and automatically, at latest when closing the edit window. However, you may use the Save command to ensure the buffer is flushed at a given time.

The in-place edit mode is preferable if the data transfer from the original to the temporary file and vice-versa, which is obligatory in default edit mode for certain operations, consumed too much time or disk space. This may be the case when opening very large files or when modifying huge amounts of data. Since usually no temporary files are needed in in-place edit mode, this edit mode is generally faster than the default edit mode. The in-place edit mode is the only mode available when using the memory editor. Hint: Even in in-place edit mode the creation of a temporary file is unavoidable when altering the file size.

If you open files using the operating system (e.g. via File | Open, from any drive letter currently available in Windows), then operating system file write commands will be used to change a file. However, in WinHex it is even possible to edit files without using operating system file write commands, directly on a disk/in a raw disk image in any file system supported, even if not supported by Windows, even files not seen by Windows (e.g. deleted files), even in partitions not seen by Windows (e.g. by damaged or deleted), without changing any timestamps or attributes, in in-place mode only. For this editing capability, the file must been opened from within the already opened volume that contains it, via the Open command in the directory browser context menu or in File mode (forensic license only). Compressed files or generally files within other files (e.g. e-mails and attachments in e-mail archives) cannot be edited, except in an evidence file container if they have been copied there from the original disk/image.

Note that files cannot be shortened or expanded that way, only the data in already allocated areas can be modified. Editing files opened directly from within disks/raw images as described above is possible in WinHex only, not in X-Ways Forensics or X-Ways Investigator, where sector level write access (to which file editing is internally translated) is disabled and where the only mode available for disks and interpreted images and files opened from within volumes continues to be read-only mode.

In forensic computing, electronic discovery and IT security, this editing capability can be helpful to manually redact (e.g. overtype) specific data that should not be examined/disclosed/seen or to securely erase specific areas within files (e.g. define as a block and fill the block). Note that evidence file containers are raw images if they have not been converted to the .e01 evidence file format and thus allow for retroactive file editing, which, however will invalidate any accompanying hash values. It is even possible to edit directories, i.e. the clusters with directory data, e.g. INDX buffers in NTFS, for example if you need to redact the names of certain files.

7.2 Scripts

Some of the functionality of WinHex can be used in an automated way, e.g. to speed up recurring routine tasks or to perform certain tasks on unattended remote computers. The ability to execute scripts other than the supplied sample scripts is limited to owners of professional licenses or

higher. Scripts can be run from the Start Center or the command line. While a script is executed, you may press Esc to abort. Because of their superior possibilities, scripts supersede routines, which were the only method of automation in previous versions of WinHex.

WinHex scripts are text files with the filename extension ".whs". They can be edited using any text editor and simply consist of a sequence of commands. It is recommended to enter one command per line only, for reasons of visual clarity. Depending on the command, you may need to specify parameters next to a command. Most commands affect the file or disk presented in the currently active window.

See Appendix B for a description of currently supported script commands.

7.3 X-Tensions API

Automate investigative tasks and extend the functionality of X-Ways Forensics with *X-Tensions*: The new X-Ways Forensics X-Tension API (application programming interface) allows you to use many of the advanced capabilities of the X-Ways Forensics computer software programmatically and extend them with your own functionality. For example, you could implement some specialized file carving for certain file types, automated triage functionality, alternative report generation, or automatically filter out unwanted search hits depending on your requirements etc.

Among other things, X-Tensions allow you to:
- read from a disk/partition/volume/image
- retrieve abundant information about each file and directory in the volume snapshot
- read from any file
- create new objects in the volume snapshot
- assign files to report tables
- add comments to files
- process, validate and delete search hits
- and do practically *everything else that is possible with a Windows program*! (thanks to the Windows API)

You can use your programming language of choice, e.g. C++, Delphi, or Visual Basic, and do not have to learn any new programming language. You can use your compiler of choice, for example Visual Studio Express (freeware).

Since an extension is not an interpreted script, but regular compiled executable code that is running in the address space of the application itself, you can expect highest performance, the same as with internally implemented functionality. X-Tensions give you easy and direct access to crucial and powerful functions deep inside X-Ways Forensics.

When X-Tensions functions can get called:
- when refining the volume snapshot
- when running a simultaneous search
- via the directory browser context menu

- in future versions of X-Ways Forensics via the search hit context menu

The X-Tension API also llows the development and use of so-called Disk I/O X-Tensions. These are snap-ins that sit between all analysis functionality and the user interface of X-Ways Forensics on the one hand and a disk/image/RAID/partition/volume from which sectors are read on the other hand. They can for example deal with full disk encryption and decrypt the data in all sectors read by X-Ways Forensics on the fly when needed, so that all relevant functions only get to see the decrypted data and can deal with it as if it was a normal disk/volume.

The user may open a selected evidence object through such a Disk I/O X-Tension using a new command in the context menu of the Case Data window. After selecting the intended X-Tension DLL, if the DLL signals that it can successfully deal with the data in that evidence object, the case will remember which DLL that was chosen and automatically apply it next time when opening the same evidence object. Note that as always partitions count as evidence objects themselves. That way full disk encryption can be tackled as well as volume level encryption.

You may distribute your XWF extension DLLs that you compile and/or your source code free of charge or even for a fee, under whatever license terms you see fit.

For more information please see http://www.x-ways.net/forensics/x-tensions/api.html.

7.4 WinHex API

The WinHex API (application programming interface) allows to use the advanced capabilities of the WinHex Hex Editor programmatically from your own C++, Delphi, or Visual Basic programs. In particular, it provides a convenient and simple interface for random access to files and disks.

Developing software that uses the WinHex API requires a valid *professional* or *specialist* WinHex license. Additionally, you need import declarations for your programming language of choice, the library file “whxapi.dll”, and the API documentation. Please find those files and more detailed information online at http://www.x-ways.net/winhex/api/.

You may also *distribute* both any software that makes use of the WinHex API and WinHex itself. There are two ways how to distribute WinHex:

1. Distribute the unlicensed WinHex version. For the API to work, your customer has to purchase professional or specialist licenses according to the number of WinHex installations needed.
-or-
2. Recommended: distribute a special API version of WinHex that is configured to only provide the API functionality and that is available at a reduced price. You may place your order online at http://www.x-ways.net/winhex/api/. Volume discount available on request (please specify the number of licenses you are interested in). One WinHex API license needed per end user computer. The product will be licensed to you, you will be the actual owner of the licenses, but any of your customers may use them. The end user does not have to take care of anything related to WinHex.

7.5 Disk Editor

The disk editor, that is part of the Tools menu, allows you to access floppy and hard disks below the file-system level. Disks consist of sectors (commonly units of 512 bytes). You may access a disk either logically (i.e. controlled by the operating system) or physically (controlled by the BIOS). On most computer systems you can even access CD-ROM and DVD media. There is an optional raw mode for optical drives that allows to read from audio CDs and also the complete 2352-byte sectors on data CDs (CD-ROM and Video CDs) that contain error correction codes.

Opening a *logical drive* means opening a contiguous formatted part of a disk (a partition) that is accessible under Windows as a drive letter. It's also called a "volume". WinHex relies on Windows being able to access the drive. Opening a *physical disk* means opening the entire medium, as it is attached to the computer, e.g. a hard disk including *all* partitions. It could also called the "raw device". The disk normally does not need to be properly formatted in order to open it that way.

Usually it is preferable to open a logical drive instead of a physical disk, because more features are provided in this case. For example, "clusters" are defined by the file system, the allocation of clusters to files (and vice versa) is known to WinHex, "free space" and "slack space" have a meaning. If you need to edit sectors outside a logical drive (e.g. the master boot record), if you wish to search something on several partitions of a hard disk at the same time, or if a partition is damaged or formatted with a file system unknown to Windows, so Windows is unable to make it accessible as a drive letter, you would open the physical disk instead. From the window that represents a physical medium you can usually also open individual partitions, by double-clicking them in the directory browser of that window. WinHex understands conventional MBR partitioning, GPT (GUID partition type), Apple partitioning, superfloppy format, Windows dynamic disks as organized by the LDM (Logical Disk Manager, MBR and GPT style), LVM2 (MBR and GPT style), and PC-compatible BSD disklabel. All dynamic volume types are supported: simple, spanned, striped, and RAID 5. Holding the Ctrl key when opening hard disks disables detection and special handling of dynamic volumes and ensures the hard disk is treated like it has been partitioned in the conventional way. Some of the aforementioned partitioning types are supported with specialist and forensic licenses only.

Please note the following limitations:

- Administrator rights are needed to access sectors on any kind of media. Under Windows Vista/7/8 you need to run the program as administrator specifically, just being logged on as administrator is *not* sufficient.
- Remote (network) drives cannot be accessed sector-wise.
- X-Ways Forensics cannot edit disk sectors or sectors in interpreted images at all, only WinHex can.
- WinHex cannot write to CD-ROM or DVD.
- Under Windows Vista/7/8, WinHex cannot write sectors on the partition with the active Windows installation and on the partition where WinHex is running from.

The appendix C of this manual provides you with specifications of the master boot record, which can be edited using the disk editor.

Save Sectors: To be used analogously to the Save command for files. Part of the File menu. Writes all modifications to the disk. Please note that, depending on your changes, this may severely damage the integrity of the disk data. If the corresponding undo option is enabled, a backup of the concerned sectors is created, before they are overwritten. *This command is only available in the full version.*

7.6 Memory Editor/Analysis

The memory editor allows to examine the physical RAM/main memory and the logical memory of a process (i.e. a program that is being executed) in a live system. All memory pages committed by a process are presented in a continuous block. Unused (free or reserved) pages are ignored by default, but optionally included and displayed with "?" characters. With no such gaps, you may compare memory dumps to files exactly with one another (absolute and virtual addresses are identical), e.g. to examine stack and heap states or observe virusses.

If you expand one of the listed processes in the list, you may open either the so-called primary memory or the entire memory of this process or one of the loaded modules (DLLs). The primary memory is the lower part of the address range, below the area where system DLLs are loaded. Usually it also contains the main module of a process (the EXE file), the stack, and the heap. The "entire memory" contains all the allocated pages in the entire logical memory address space of a process.

With the 64-bit edition of WinHex/X-Ways Forensics you can get loaded modules above the 4 GB barrier in 64-bit processes listed, and read and edit memory in such address ranges. Unicode is supported for process and module names and paths in the memory editor. Page boundaries are represented by horizontal lines. Boundaries that represent gaps between contiguous allocated regions are represented by darker horizontal lines. The Info Pane shows information such as the maximum address represented and the number of allocation gaps (=number of contiguous allocated page ranges -1) as well as protection status and type of the currently displayed page.

Please note the following limitations:

- Access to *physical* RAM under Windows XP (32-bit) only, no more than 4 GB, and with administrator rights only
- Caution: Only keyboard input can be undone!
- Editing is possible in in-place mode only.
- The evaluation version only supports view mode.

The options relevant for the memory editor are "Check for virtual memory alteration" and "Virtual Addresses".

Main Memory Analysis

Requires a forensic license. When you open the local physical RAM (via Tools | Open RAM, only under Windows XP) or a main memory dump as a file (and interpret that file exactly like you would a disk image) or add a memory dump to a case, processes will be listed in the

directory browser, even hidden processes, with their timestamps and process IDs, and their own respective memory address spaces can be individually viewed in "Process" mode, with pages concatenated in correct logical order as seen by each process. The "particularly thorough data structure search" is signature-based, will take a little longer than taking a standard volume snapshot and may turn up traces of additional processes including rootkits. Memory can be acquired remotely with the help of F-Response (Tools | Open Disk). The analysis is supported for most (but not all) variants (service packs) of Windows 2000, Windows XP, Windows 2003 Server, Windows Vista, Windows 2008 Server, and Windows 7, 32 bit and (less complete) 64 bit. Only complete memory dumps are supported, those which include regions in RAM that are utilized by the BIOS and by PCI devices.

Windows kernel data structures and named objects are conveniently listed in a tree in the volume snapshot under “Objects”. Loaded modules are listed under “Modules”. That enables X-Ways Forensics to allocate the memory pages in RAM mode that they occupy to them, and to compute hashes for them so that they can be identified via special hash sets. For hashing purposes it is recommended to list the invariant headers of loaded modules only (see Volume Snapshot Options).

The technical details report informs you of important system-wide parameters as well as of the current addresses of important kernel data structures and loaded kernel modules. In Details mode you can find the addresses of process-related data structures for each process and the ID of its parent process. In RAM mode, the Info Pane shows for each memory page a process to which it is allocated (if any) and its memory management status.

With the appropriate background knowledge, this functionality can be used learn more about the current state of the machine and its processes, sockets, open files, loaded drivers, and attached media, to identify malware, to find the decrypted version of encrypted data, to analyze network traces in incident response, and to do further research in the field of memory forensics.

7.7 Template Editing

A template is a dialog box that provides means for editing custom data structures in a more comfortable and error-preventing way than raw hex editing does. Editing is done is separate edit boxes. Changes take effect when pressing the **ENTER** key or when quitting the template after being prompted. The data may originate from a file, from disk sectors, or from virtual memory. Especially when editing databases, you may prefer to define a custom template for ease of access to the records. You will find the command to print a template in the system menu.

A *template definition* is stored in a text file with the extension .tpl. The *template editor* enables you to write template definitions and offers syntax checking. A template definition mainly contains variable declarations, that are similar to those in source code of programming languages. The syntax is explained in detail in Appendix A. The supported data types include all the common integer, floating-point and boolean variants, date types, hex values, binary, characters, and strings type. Arrays of both single variables and groups of variables can be used.

The ability to move freely forwards and backwards within the data makes using templates

particularly flexible:

- The same variable may be interpreted and manipulated in several ways.
- Irrelevant data sections can be skipped.

The *template manager* lists all text files in the WinHex directory that contain template definitions. The title of the template along with a description, the filename, and the date and time of the last modification are shown. Click the Apply button to display a template using the selected template definition for the data in the current editor window at the current position. You may also create a new template definition, delete or edit an existing one.

WinHex comes with several sample templates.

8 Data Recovery

8.1 File Recovery with the Directory Browser

Most obviously, deleted files and directories that are listed in the directory browser can be recovered easily and selectively with the directory browser's context menu. You navigate to a directory (or explore the root directory recursively), select the files to recover, and use the Recover/Copy command in the context menu. See chapter "directory browser".

8.2 File Recovery by Type/File Header Signature Search

Data recovery function in the Disk Tools menu, and also a strategy to find previously existing files as part of the Refine Volume Snapshot command. This recovery method is also referred to as "file carving". It searches for files that can be recognized by a characteristic file header signature (a certain sequence of byte values). Because of this approach, file carving does not depend on the existence of functional file system structures. When found based on the signature, the files are saved to the output folder that is specified by the user (File Recovery by Type) or merely listed in a virtual directory of the volume snapshot (File header signature search). Optionally, recovered files of each type are put into their own subfolder (...\JPEG, ...\HTML, etc.). Note that file carving assumes contiguous file clusters, so produces corrupt files in case the files were originally stored in a fragmented way. A log file "File Recovery by Type.log" about the selected parameters and the recovery results is written to the output folder for verification purposes.

You can expand or collapse the entire file type tree in this dialog window with a single mouse click on the appropriate button. That is useful because when expanded you only need to type the first few characters of the file type description to automatically jump to the first matching item in the tree.

Since no use is made of a possible presence of a (consistent or damaged) file system, the original

file sizes are principally *unknown* to this recovery method, and so are the original *filenames*. That is why the resulting files are mostly named generically according to the following pattern: Prefix#####.ext. "Prefix" is an optional prefix you provide. #####" is an incrementing number per evidence object. "ext" is the filename extension that corresponds to the file header signature according to the file type definition. The output filename prefix may optionally contain a placeholder "%d", which will be replaced by the drive name. This is useful if you apply File Recovery by Type to multiple drives at a time and wish to be able to easily distinguish files from different drives.

With a specialist license or higher, the "intelligent naming" option will cause Exif JPEG files to be named after the digital camera model that created them and their internal time stamp, if available. Many Windows Registry hive files are given their original names, also some JPEG files in whose metadata Photoshop has embedded a name. JPEG files without known name and no Exif metadata that however have been created by a known generator get some additional information in their artificial name in parantheses: First a designation of the generator signature (currently e. g. "IJG Library", "Photoshop" and "Photoshop Web", where the latter stands for Photoshop when saved with optimization for usage in the web), second the quality setting "Q=" when the JPEG file was saved, which ranges from 1 to 100 or 1 to 7 in the case of Photoshop. Further generators (Apple, Canon, etc.) theoretically could also be identified, but not in the official release version of WinHex/X-Ways Forensics. Thumbs.db files are always named thumbs.db, index.dat always index.dat. The aforementioned prefix is not used in conjunction with original filenames.

Various algorithms are at work internally that try to determine the original sizes of files of many different types (among others, JPEG, GIF, PNG, BMP, TIFF, Nikon NEF, Canon CR2 raw, PSD, CDR, AVI, WAV, MOV, MPEG, MP3, MP4, 3GP, M4V, M4A, ASF, WMV, WMA, ZIP, GZIP, RAR, 7Z, TAR, MS Word, MS Excel, MS PowerPoint, RTF, PDF, HTML, XML, XSD, DTD, PST, DBX, AOL PFC, Windows Registry, index.dat, Prefetch, SPL, EVTX, EML) by examining their data structure. This applies to entries in the file type definition database that have a "~" in the Footer column. These entries should not be altered in order for the size and type detection to work for these file types. Alternatively, a footer signature can also help to find the end of a file. Files for which neither an internal algorithm nor a footer signature definition exists or file about whose original size the available internal algorithm has no idea and for which no footer signature is actually found, are recovered at the default size specified in the file type definition database in bytes. Be generous when specifying such a size because whereas files recovered "too large" can still be opened by their associated applications, prematurely truncated files often can't be as they are incomplete. The attempt to detect the original size of files of certain types by searching for a footer is limited by a *size detection limit*, which is optionally specified in the database as well, after the default size and a forward slash. Such a limit is necessary to avoid that a footer for a given file is searched within the whole volume, which would be very time-consuming if the volume is large. Also, it becomes increasingly unlikely to find the right footer if not in the immediate vicinity of the header, and even if found very far apart, such a file is likely fragmented or partially overwritten etc. The standard default size (if not specified) is 1 MB. The standard maximum size (if not specified) is 64 times the default file size.

File headers are usually found at cluster boundaries because that is where file systems mostly put the start of a file. However, it is more thorough (and not slower) to search for *sector*-aligned file headers because that allows to also find files from previously existing partitions with a different

cluster layout. If performed on a physical medium or raw file with no cluster layout defined, WinHex searches at sector boundaries anyway. There is yet another possibility, a thorough *byte*-level search. This is required when you are trying to find files that are not reliably aligned at any sector boundaries (e.g. files in backup files or tape images or embedded in other files) or when trying to find entries/records/micro-formats/memory artifacts etc., i.e. not complete ordinary files. This comes at the cost of a possibly increased number of false positives, though, misidentified file signatures occurring randomly on a media, not indicating the beginning of a file. Individual flags in the file type definition database can help on a per file type basis to decide which files to search for a cluster, sector or byte boundaries.

That the start sectors of files that are already known to the volume snapshot are always excluded from file carving is optional. Of course, X-Ways Forensics generally still tries to prevent duplicates, but if the file header signature definition or the internal file size detection is strong enough to suggest that a known deleted file was overwritten with a new file, then that new file will be carved although it shares the same start sector with the known file.

If you intentionally abort the file header signature search or if the file header signature search causes X-Ways Forensics to crash, next time when you start a file header signature search in the same evidence object, you will find an option to resume it right where it was interrupted, or where it was when the volume snapshot was last saved before the crash occurred (depends on the auto-save interval of the case).

You may limit the scope of the recovery to a currently selected block if necessary and/or to allocated or unallocated space (option available on a logical drive or volume). E.g. in order to recover files that were deleted, you select to recover from unallocated space only. Files that are not accessible any more because of file system errors may still be stored in clusters that are considered as in use.

The option "Ext2/Ext3 block logic" causes this recovery method to deviate from the standard assumption of no fragmention in that it will follow the typical Ext block pattern, where e.g. the 13th block from the header of the file is considered an indirect block that references the following data blocks. This option has no effect when applied to partitions that WinHex knows have a file system other than Ext2 and Ext3 or when a header is found that is not block-aligned.

The effects of NTFS compression on file data can optionally be compensated for in a file header signature search (forensic license only), in many cases successfully. If the signature of an NTFS-compressed file is found, the file will be marked as compressed, and an attempt will be made to decompress the file “on the fly” when needed with a sophisticated algorithm that can even decompress files that consist of multiple compression units.

8.3 File Type Definitions

“File Type Signatures *.txt” are tab-delimited text files that serves as a file type definition database for refining volume snapshots and for the File Recovery by Type command.

WinHex comes with various preset file type signatures. You may fully customize the file type

definitions and add your own ones, either in "File Type Signatures Search.txt" or in any additional such files of the same format named “File Type Signatures *.txt”, which will be loaded as well and may have the benefit that they will not be overwritten when you install the next update if they don't have the same name as one of the default files. Only if the filename contains the word “search”, the file types will be available for file header signature searches. Otherwise they are used for file type verification only of files that are already part of the volume snapshot (forensic license only). Up to 4096 entries are supported altogether (1024 for searching).

When you click the Customize button to edit the file "File Type Signatures Search.txt", by default WinHex opens the file in MS Excel. This is convenient because the file consists of columns separated by tabs. If you edit the file with a text editor, be sure to retain these tabs, as WinHex relies on their presence to properly interpret the file type definitions. MS Excel retains them automatically. After editing the file type definitions, you need to exit the dialog window and invoke the File Recovery by Type or Refine Volume Snapshot menu command again to see the changes in the file type list.

1st column: File Type

A human-readable designation of the file type, e.g. "JPEG". Everything beyond the first 19 characters is ignored.

2nd column: Extensions

One or more file type extensions typically used for this file type. E.g. "jpg;jpeg;jpe". Specify the most common extension first because that one will be used by default for naming recovered files. If that first extension is specified in upper-case characters, it will be used by the file type verification to fill the Type column for a file even if the file has one of the alternative plausible filename extensions. More than 255 characters supported.

3rd column: Header

A unique header signature by which files of this file type can be recognized. It is specified in GREP syntax (see Search Options for an explanation), so that it's possible to match variable byte values (e.g. [\xE1\xE2] mean "the byte value could be 0xE1 or 0xE2") or undefined areas (.).The maximum length of the represented signature is 48 bytes. To find out characteristic file header signatures in the first place, open several existing files of a certain type in WinHex and look for common byte values near the beginning of the file at identical offsets.

4th column: Offset

The relative offset within a file at which the signature occurs. Often simply 0. The signature must be contained in the first 512 bytes.

5th column: Footer

Optional. A signature (byte sequence) that reliably indicates the end of a file, specified in GREP syntax. GREP expressions that represent variably-sized data may not work as expected. A footer signature may help to achieve a recovery with the correct file size. The recovery algorithm does

not search for the footer further than the number of bytes specified as the maximum file size, starting from the header.

Even better than a footer is the potential availability of an internally implemented algorithm in X-Ways Forensics that knows the file format well and can usually find out the correct file size if a file is not fragmented, incomplete or corrupt. Such an algorithm is indicated in the Footer column with a tilde (~) and an algorithm ID number.

6th column: Default size

Optional. 1 or 2 values. If 2 values, the second one is a file type specific size detection limit and delimited from the default size by a forward slash.

7th column: Flags

Optional. Can further tailor file carving for certain file types and are yet another indicator of how sophisticated and powerful file carving is in X-Ways Forensics.

b (lower case): The signature is searched at the byte level when given the choice. Useful especially for entries/record/micro-formats/memory artifacts (i.e. not complete ordinary files) that are not typically aligned at any sector or cluster boundaries.

B (upper case): Prevents a byte-level search for that particular signature, for performance reasons.

c (lower case): If taken into account (depends on user interface settings), ignores header signatures that are not aligned at cluster boundaries. Can be useful for some file types to avoid to many false positives.

C (upper case): Denotes file type signatures that should not be used to search for NTFS-compressed files if compensation for NTFS compression is active, because they are too weak and would yield too many false positives or would not be actually stored as compressed anyway.

u (lower case): Stands for "unused". Allows to carve files only in clusters that are free according to the file system.

U (upper case): Allows to carve files only in clusters that are free according to the file system and also not used by previously existing files as contained in the volume snapshot.

f (lower case): Indicates that the specified footer signature is used to find data that is not part of the file any more and should excluded. Ordinary footers are included in the carved file. Useful for file formats that do not have a well defined footer, where the end of the file can be detected by the occurrence of data that does not belong to the file any more. That could be the same signature as the header (if files of that type occur typically in groups, back to back) or just \x00 (for file formats such as text files that do not contain zero-value bytes, where however \x00 can be expected with a high likelihood in the RAM slack). Such footer signatures should be marked as exclusive because the data matched by it is not part of the file itself.

F (upper case): Makes X-Ways Forensics discard hits of the file header signature search if no corresponding footer can be found, provided that a footer signature is specified in the definition. Can be useful to reduce the number of or totally avoid false positives.

h: Indicates that the specified *h*eader signature is used to find data that is not part of the file itself. That means that the header signature will be excluded from the carved file. The carved file will start after the header signature.

G: Stands for “greedy”. Greedily allocates all the sectors exclusively. The file type signature search continues its search for further file headers only after the presumed end of such files. Can be useful if an internally implemented algorithm is available that is sure that the carved file contains all valid data, so that it is not necessary to search for other files within the previously carved file's boundaries.

g (lower case): Weaker version of the same flag. Only if an internal file size detection algorithm exists for a file type and if a file with the same start sector number exists already with the same file size as detected, the "g" flag will cause X-Ways Forensics to skip the affected sectors. This can help to prevent overlapping zip files and thereby avoid potentially many contained duplicate files.

S: Marks signatures that are good enough for the file header signature search (probably in conjunction with a carving algorithm), but not for file type verification because of occasional misidentifications. This flag should be very rarely needed.

t: Prevents X-Ways Forensics from presenting the type of carved files immediately as confirmed. Useful for example for file format families such as XML, to determine the exact subtype later during file type verification.

e: Stands for "embedded". If a file type has a tilde (~) algorithm in the Footer column and is marked with this flag, it will be searched embedded in certain other files during volume snapshot refinements, always at the byte level.

E: Never carved as an embedded file within other files.

W (upper case): Identifies header signatures that are too weak to newly detect the type of a file and are merely used to confirm the type suggested by the name extension of the file.

8.4 Manual Data Recovery

It is possible to restore lost or logically deleted files (or more general: data) that are merely marked as deleted in the file system, but have not been *physically* erased (or overwritten).

Open the logical drive where the deleted file resided on using the disk editor. Principally you can recreate such a file by selecting the disk sectors, that were allocated to the file, as the current block and saving them using the menu command Edit | Copy Block | Into New File. But it may prove difficult to *find* the sectors where the file is still stored. There are two general ways to

accomplish this:

1. In case you know a snippet of the file you are looking for (e.g. the characteristic signature in the header of a JPEG file or the words “Dear Mr. Smith” in a MS Word document), search it on the disk using the common search commands (“Find Text” or “Find Hex Values”). This is a very simple and safe way, and can be recommended to anyone.

2. In case you only know the filename, you will need some knowledge about the filesystem on the disk (FAT16, FAT32, NTFS, ...) to find traces of former directory entries of the file and thereby determine the number of the first cluster that was allocated to the file. Detailed information on file systems is available on the WinHex web site. The following applies to all FAT variants:

   If the directory that *contained* the file (let's call that directory “D”) still exists, you can find D on the disk using Tools | Disk Tools | List Directory Clusters. The factory template for FAT directory entries that comes with WinHex will then be helpful to find out the number of the first cluster that was allocated to the deleted file in that directory. Otherwise, if D has been deleted as well, you need to find the contents of D (using the directory entry template) starting with the directory that contained D.

   Deleted files and directories are marked with the character “å” (hexadecimal: E5) as the first letter in their name.

You may encounter the problem that the file to recover is fragmented, i. e. not stored in subsequent contiguous clusters. On FAT drives, the next cluster of a file can be looked up in the file allocation table at the beginning of the drive (simple templates to do this can be found on the web site), but this information is erased when a file is deleted.

9 Options

9.1 General Options

1st column:

- Under Windows Vista and 7 it may be recommendable to **always run** WinHex/X-Ways Forensics **as administrator** if you need sector-level access to media.

- The option **Allow multiple program instances** allows you to execute WinHex more than once at a time. If not checked, WinHex makes the main window of the previous instance the foreground window instead of creating a new program instance. By default, this option is half selected. That means you will be given a choice when executing the .exe file again, whether to start a new instance or not. At that time you may also try to recover a previous instance if caught in an infinite loop.

- At startup, WinHex can optionally **show** the **Start Center** or **restore** the **last window arrangement** (all windows with their sizes and the positions as you left them in the precedent WinHex session).

- By default, **edit windows** are not **opened** in a **maximized** state.

- Specify the number of **recent**ly opened **documents** to remember and to **list** in the Start Center (255 at max.). Up to 9 of them are also listed at the end of the File menu.

- **Do not update file time** means that WinHex will preserve the last modification time when a modified file is saved with File | Save or Save As.

- **More context menus**: If fully checked or if the Shift key is pressed while right-clicking a directory in the Case Data window, a context menu appears that allows to recursively explore the right-clicked directory (just like when no context menu is shown), allows to tag the directory recursively (just like when pressing the Space bar), to expand the directory recursively (just like when pressing the multiply key of the numeric keypad), to collapse all, export a subtree into an ASCII text file, or copy the entire path of that directory into the clipboard. If at least half checked or if the Shift key is pressed while right-clicking the hex editor display, a suitable context menu will appear there as well.

- You may have **WinHex** appear in the Windows **context menu**. The shell displays the context menu when the user clicks an object with the right mouse button. WinHex provides menu items for files, folders and disks. If this option is not fully selected, there is no menu item for files.

- **Save program settings in .cfg file:** If half checked, the settings are saved whenever the program terminates (cleanly). If fully checked, then every time when you click OK in any dialog window (could be useful if the program does not terminate cleanly, to avoid that you lose your latest settings). If totally unchecked, the program settings will not be saved at all, except if you hold the Shift key when exiting the program, which is necessary once if you would like to save in the .cfg file the setting that from then on the settings should not be saved again.

- By default WinHex **numbers** disk **partitions** in the order of their physical **location**.

- If **Auto-detect deleted partitions is enabled**, WinHex tries to identify obvious deleted partitions automatically in gaps between existing partitions and in unpartitioned space directly following the last partition, when opening physical hard disks. Such additionally detected partitions will be listed in the Access button menu and marked as deleted. Please note that deleted partitions detected in gaps between existing partitions cause the partition numbering to be changed. E.g. an existing partition #3 might become partition #4 if a deleted partition is detected on the disk before it.

- The **Sector reading cache** accelerates sequential disk access by the disk editor. This option is recommended particularly when scrolling through CD-ROM and floppy disk sectors, since the number of necessary physical accesses is significantly reduced.

- If **Check for surplus sectors** is disabled, WinHex will not try to access surplus sectors when a physical hard disk is opened. When additional sectors are detected, WinHex will remember them the next time you open the disk. You may enforce a new check by holding the Shift key while opening the disk. Checking for surplus sectors may cause very long delays, strange behavior or even damage to the Windows installation on *some very few* systems.

- The **alternative access method 1** for physical hard disks may allow to access hard disks formatted with an unconventional sector size or other media that cannot be accessed otherwise. Note that it may be slower than the regular access method. If considerably slower, WinHex will notify you of this and recommend to revert to the standard access method. **Access method 2** affects physical hard disks only as well. Both alternative methods allow you to specify a timeout in milliseconds after which read attempts will be aborted. This can be useful on disks with bad sectors, where an attempted read access to a single sector could otherwise cause a delay of many seconds or minutes.

- The **substitute pattern for unreadable sectors** is always used instead of the original data stored in disk sectors if these sectors cannot be read, for all purposes (display on the screen, imaging, cloning, hashing, searching, ...). If you are going to hash disks with bad sectors and want to compare/reproduce the results with other tools, then you can specify the same pattern as used by the other tool here. Just note that such hash values are difficult to reproduce because bad sectors could multiply in the course of several attempts. If when trying to read bad sectors you prefer to get zero-value bytes delivered back, totally remove the pattern (ensure that the edit box is completely blank).

2nd column:

- Specify the **folder** in which to create **temporary files**. By default that is the directory indicated by the TEMP variable in your Windows system. Instead of an absolute path you may also specify a dot (.) as a placeholder for the directory from where WinHex/X-Ways Forensics is executed. Or .. for the parent directory of that directory. Or partial path relative to either the . or .. directory (e.g. .\temp or ..\temp). This concept applies also to the next four folders.

- Specify the **folder** in which to create and expect **images and backup files** (.whx).

- Specify the **folder** in which **cases and projects** are created and expected.

- Specify the **folder** in which **templates and scripts** are stored.

- Specify the **folder** in which to maintain the **internal hash database**. The hash database of block hash values, if used at all, is stored in a directory at the same level, with the same base name plus " [block hash values]" appended.

In all of these standard paths you may use system and user environment variables, where the variable name has to be enclosed in percentage signs, e.g. %TEMP%.

- **X-Ways Investigator [CTR]/X-Ways Imager GUI:** Available when operated with a forensic

license. Allows to activate the considerably reduced user interface of X-Ways Investigator [CTR], which is meant for investigators
- who are specialized in a certain area e.g. of white-collar crime
- who do not need profound knowledge of computer forensics
- who do not need technical insights that WinHex and XWF are well-known to offer
- who receive e.g. convenient-to-handle X-Ways evidence file containers from well-versed computer forensics examiners with only selected files from various sources (e.g. "all documents that contain the keywords x and y"), with obviously irrelevant stuff already filtered out
- who need to review hundreds of electronic documents, identify relevant ones, add comments to them, identify logical structures and connections between them with the help of their comments, and print documents, all within the same environment with a few mouse clicks, which saves the time to extract and load each document in its associated application
- who may or may not need to work in an environment severely restricted by the system administrator anyway

The X-Ways Investigator interface lacks many advanced technical options, to allow for easier access to non-technical personnel. X-Ways Investigator licenses that only allow to use this GUI are available at 50% the regular rate on request. An optional file "investigator.ini" controls additional simplifications and administrative security precautions, e.g. to allow users to open evidence file containers only, and only such containers that have been classified as secure.

- If you select **Show file icons**, the icons stored in a file are shown in the info pane. If a file contains no icons, the icon of the file *type* is shown if this option is “fully” selected. Only for files opened with the File | Open menu command.

- Last not least, you may select one of several different dialog window and button styles.

- With a forensic license, you may monitor lengthy operations from other computers in the same network, i.e. see whether they are still ongoing or completed. You can enable **progress notifications** via text files (that can be created in a directory on a network drive) and via e-mail, in user-defined intervals. Multiple recipient e-mail addresses can be specified as well if delimited by commas. The correct SMTP port is often 25, sometimes 587. The correct settings are provided by your administrator or Internet provider.

3rd column:

- The **ENTER** key can be used to enter up to four two-digit hex values. A useful example is **0x0D0A**, which is interpreted as an end-of-line marker in the Windows world (Unix: 0x0D). The Start Center could then still be opened using **SHIFT**+**ENTER**.

- Decide whether you want to use the **TAB** key to switch from text to hexadecimal mode and vice versa or to enter the TAB character (0x09). In any case, **TAB**+**SHIFT** can be pressed to switch the current mode.

- Non-printable **characters** with a character set value smaller than **0x20** can be represented by a user-defined other character.

- The **bytes** in the **display** can be represented **as** characters in the **text** column **one by one**, or WinHex can try to combine them, which if the active code page in Windows is a double-byte character set *may* be desirable to get the characters right (if 2 bytes = 1 character), or undesirable because of the variable row length. This has an effect only if View | Character Set | * ASCII is selected, as only then the code page active in Windows can make a difference for the display.

- **Offsets** can be presented and prompted for in a decimal or **hexadecimal** notation. This setting is valid for the entire program.

- When using the **memory editor**, it may be useful to have WinHex display **logical** memory **addresses** for processes instead of zero-based, linear, contiguously counted offsets. This is always done in hexadecimal notation. The dialog window of the Goto Offset command will also prompt for logical addresses.

- **Page** and sector **separators** may be **displayed**. If this option is enabled partially, only sector separators are displayed.

- Specify the number of **bytes per line** in an edit window. Common values are 16 or 32 (depending on the screen resolution).

- Choose how many **bytes** shall be displayed in a **group**. Powers of 2 serve best for most purposes.

- There is an option to define the size of the extra gap between rows in the hex editor display in pixels, which together with the official height of the selected font defined the distance between the rows. The default value has always been 3 before v17.2, but now it can be decreased, to display more rows at the same time and see more data. For example with the Courier font the display still looks fine with an extra gap of 1, but you see 15% more data (based on font size 10). Even negative values are possible. With -1 you may see 35% more data than before.

- **Search hit highlighting in File mode**: Option to get all search hits in a file highlighted in File mode at the same time, either only when a search hit list is displayed (if half checked) or permanently once search hits have been loaded for an evidence object, i.e. even when working with the normal directory browser (if fully checked). Search hits are loaded after an evidence object has been opened as soon as search hits are listed. This feature also applies to user search hits. Requires forensic license.

- **NTFS: MFT auto coloring:** Highlights the various elements in FILE records of the NTFS file system, when the cursor is located within such a record, to facilitate navigation and understanding. Requires a specialist or forensic license. Also automatic highlighting of aligned FILETIME values in Disk/Partition/Volume and File mode is available. Useful when manually inspecting files of various Microsoft formats which may contain more timestamps than can be automatically extracted (try e.g. with index.dat, registry hives, .lnk shortcut files etc.). If the lower half of a data window has the focus and FILETIME values are highlighted,

you may also hover the mouse cursor over such a value to get a human readable interpretation of the timestamp. Alternatively, of course, you could get it from the data interpreter if you click the first byte of the value.

- Select a **color** used as the **background** of the current **block**. You can only change the color if the option "Use Windows default colors" is switched off.

- Select a **color** used as the **background** of every other fixed-length **record**, if record presentation is enabled (see Position menu).

- Select the default **color** for newly created **annotations**/positions/bookmarks.

- You may want WinHex to **highlight modified bytes**, i.e. display altered parts of a file, disk, or memory in a different color, so you can distinguish between original data and changes you have made so far. You may select the hilite color.

- You may choose a **font** for the hex editor display, and decide whether the standard Windows GUI font should be used for the other parts of the WinHex/X-Ways Forensics GUI (via an additional checkbox).

Notation Options

- Choose your preferred date, time, and number notation settings. This is important especially to be independent of the Windows regional settings of live system that you want to preview if you are using X-Ways Forensics on a computer that is not your own one. You may also choose to display years in dates with 2 digits only.

- There is an option to output dates in the directory browser and in some other parts of the user interface in a nicer, longer and more locale-specific notation, which can include the weekday and the name of the month based in your language or in English. Also, that format is Unicode-capable, which allows for example for original Chinese notation of dates. Please see http://msdn.microsoft.com/en-us/library/dd317787%28v=vs.85%29.aspx for a complete explanation of what kind of notation is possible. Examples of how to represent the month (in English): MMMM = April, MMM = Apr, MM = 04, M = 4. Example of a complete format: d/MMM/yyyy (ddd) = 2/Apr/2014 (Wed).

- There is an option to display timestamps with a precision of milli**seconds**. You may specify the number of **digits after** the **decimal** point (up to 3). Useful for the file systems NTFS, Reiser4 and FAT, which provide for a higher precision than seconds in all or some timestamps.

- Optionally, the actually used **time zone conversion bias**, including daylight saving where appropriate, can be displayed right in the timestamp columns in the directory browser.

- **File sizes** can optionally **always** be **display**ed **in bytes** instead of rounded. If the checkbox is half checked, that applies to items in volumes only, otherwise also items on physical, partitioned media.

Factory settings of *all* options can be restored using the Initialize command of the Help menu.

9.2 Volume Snapshot Options

These options can be reached via the Directory Browser Options. Most of them take effect when taking a new volume snapshot.

- With the option **Keep volume snapshots between sessions** enabled, all information on file systems in opened volumes collected by WinHex (Disk Tools menu and/or Specialist menu) remains in the folder for temporary files even when WinHex terminates. WinHex can then reuse the snapshots in later sessions. Volume snapshots of evidence objects in a *case* are always kept, regardless of this setting, in that evidence object's metadata subdirectory.

- **Quick snapshots without cluster allocation** speeds up taking a volume snapshot (in particular for the file systems Ext2, Ext3 and ReiserFS, and in particular also when the volume snapshot files are created across a slow USB 1.1 interface or network), however, causes WinHex to lose its ability to tell each sector's and cluster's allocation (for which file it is used). You may use the command "Take New Volume Snapshot" of the Tools menu to update the view of a volume, e.g. after unchecking this option.

- **Inherit deleted state**: Causes deleted partitions to pass on their deleted state to everything that they contain (files and directories), and deleted e-mail archives to pass on their deleted state to all the e-mails, directories and attachments that they contain. This may seem logical, but results in a loss of information, as depending on the reference *everything* may be listed as deleted, even files/e-mails that from the point of the file system/the e-mail archive still existed when the partition/file was deleted. By default, this option is not selected, so that X-Ways Forensics distinguishes between existing and deleted files and e-mails etc. even in deleted partitions/deleted e-mail archives, so that more information is retained.

- **Net free space computation**: Allows you to work with an adjusted virtual free space file that is net of clusters that were identified as belonging to previously existing files, to minimize the amount of space in file systems that is read *twice* for logical searches and indexing. After changing this option or after discovery of more previously existing files, the virtual free space file is updated when it is opened next time, for example selected in File mode or when it is that file's turn during a logical search. Relative offsets of search hits in this virtual file may become wrong when it changes (for example when some more clusters are allocated to more identified previously existing files, so that the net free space file becomes smaller), so they cannot be used to navigate to the search hits in File mode. Only physical offsets of search hits, usable in Partition/Volume mode, are guaranteed to remain valid. The virtual free space will be frozen and not change any more once it has been indexed, or once it gets child objects, i.e. usually files that have been carved within it manually in File mode, because those depend on unchanged relative offsets within the virtual free space file.

- **Keep more data** of the volume snapshot **in memory**, e.g. for much quicker sorting by timestamps.

- You can indicate whether you are interested in getting files included in the volume snapshot whose clusters (and therefore data) are totally unknown, with only metadata (e.g. just filename and path), in Ext*, XFS and Reiser*. If fully checked, all previously existing files of which metadata only is known will be included in a volume snapshot. If not checked at all, those files will be ignored. If half checked, only files for which more than just the name is known (e.g. timestamps) will be included, but not directory entry remnants in Ext* or Reiser file systems.

- Extended attributes in NTFS are optionally included in the volume snapshot as child objects of the directory or file to which they belong, with the name "$EA" and marked in the Attr. column with "($EA)". Either all such attributes (if the box is fully checked) or only non-resident ones (if half-checked, default). If none at all, the clusters that belong to non-resident extended attributes of existing objects will be covered by the virtual file "misc non-resident attributes" as before. Background information: Microsoft uses extended attributes on system binaries as part of the secure boot components. Attackers have been using large extended attributes to hide malware in some high profile cases. Large extended attributes are flagged automatically by report table associations.

- **Including** logged utility streams (**LUS**) in **NTFS** in newly taken volume **snapshots** is optional. Either *all* LUS can be included (if fully checked) or only non-$EFS LUS (if half checked) or no LUS at all. Useful for NTFS volumes written by Windows Vista, if you are not interested in $TXF_DATA LUS.

- Downloaded files in NTFS can be conveniently recognized if their alternative data stream "Zone.Identifier" is represented as a report table association instead of as a child object in the volume snapshot. That means you do not need to navigate to the child object to find out what the child object might be. "ZoneId=3" as the name of the report table identifies files downloaded from the Internet.

- If you get read errors on a CD/DVD (e.g. because of scratches on the surface) when the volume snapshot is taken, you know that not all sectors with the data structures of the file system are readable. **List**ing the **ISO9660** file system's directory tree on CDs *in addition* to a possibly also existing **Joliet** file system can be useful because that means a second chance to get all directories and files listed, if the corresponding data structures of the same directories are located in *readable* sectors in the ISO9660 area.

- For better results when matching hash values against special hash sets, only the invariable **header** of **loaded modules** can be listed in main memory analysis.

- Optionally, **files** on the logical drive letters A: through Z: can be **opened** from within the directory browser with the help of the **operating system** instead of with the built-in logic at the sector level. Please note that this is forensically sound only for write-protected media. On writeable media, Microsoft Windows may update (i.e. alter, falsify) the last access timestamp of files you open. The benefit, however, is that access to such files will be noticeably faster in many situations, especially on slow media such as CDs and DVDs, e.g. when you compute hashes or skin color percentages for files in a volume snapshot, because Microsoft Windows employs read-ahead mechanisms and entertains a file caching system. Another benefit is that

files opened with the help of the operating system are editable in WinHex. Limitation: Files on multi-sessions CDs and DVDs cannot be read that way.

9.3 Viewer Programs

Here you may enable the separate viewer component and specify the path where it is located (by default: subdirectory "viewer"). The path may be relative to the directory where X-Ways Forensics is executed (.), e.g. ".\viewer" or relative to the parent directory of that directory, e.g. "..\viewer". You may decide to use it for pictures, too (instead of the internal graphics viewing library). You may select your preferred text editor and HTML viewing program. The HTML viewer program can be e.g. MS Word or NVU, i.e. a program that can be used to further edit the HTML case reports the X-Ways Forensics can create automatically. For merely viewing and printing we recommend Internet Explorer.

If the internal graphics viewing library is used to view pictures, not the viewer component, then optionally the picture viewer window can be closed automatically when a new picture is viewed.

You can also specify the .exe path of MPlayer (tested with v1.0rc2, non-GUI version, also download the separate codecs package and extract it into the "codecs" subdirectory of MPlayer) or Forensic Framer, two programs that allow X-Ways Forensics to extract pictures from videos. If mplayer.exe is found in a subdirectory \MPlayer of the installation directory of X-Ways Forensics, it will defined as the video extraction program and as an external viewer program automatically. Relative paths started with .\ or ..\ are supported, where . stands for the directory from which X-Ways Forensics is executed and .. its parent directory. Please note that we cannot provide support for external programs.

You may also specify up to 32 custom viewer programs that can be conveniently invoked from inside X-Ways Forensics via the directory browser context menu. Also you may specify which file types you prefer to view in the program that is associated with their extension in your system, typically file types that the separate viewer component does not support. There is a checkbox labelled "Append type as extension if newly identified" checkbox. Allows to more easily get Windows to run the right program for misnamed files, files without extension etc. The paths of these external viewer programs are defined in a separate file, named Programs.txt, so that it is easy to share a collection of external programs separately, or keep them when taking over all other settings from someone else. In that text file you may also change absolute paths to relative paths (with . and ..), for programs that are as portable as X-Ways Forensics itself and that you wish take along on a USB stick for analyses of live systems.

An alternative e-mail representation is available in Preview mode (also in the case report). Attachments are not linked directly from this kind of e-mail representation (yet). If this option is half selected, the e-mail header of .eml files is excluded in Preview mode (not Raw mode). Useful if you would like to see more of the body of the e-mail without scrolling. You can see subject, sender, recipient and dates already in the directory browser, and attachments are listed when exploring the parent .eml file.

Crash-safe text decoding: If enabled, text extraction from certain file types for logical searches

and indexing will be done by the viewer component in a separate process, such that if the viewer component crashes or becomes unstable, it does not render the main process (X-Ways Forensics) unstable or cause it to crash.

Buffer decoded text for context preview: If enabled, the result of the text extraction from certain file types for logical searches and indexing will be stored by X-Ways Forensics in the volume snapshot for reuse when searching/indexing again, to save time.

If the crash-safe decoding option is active, you may use a different version of the viewer component for decoding than for viewing files, at the same time. You can specify separate directories. This is useful to benefit from the extended file format support of the latest version 8.5 and at the same time employ the more reliable text decoding capabilities of the previous version 8.4.1 for PDF files produced by the OCR software Abbyy Fine Reader 11 and possibly others.

Gallery Options

- If the creation of thumbnails for **pictures within** large (e.g. solid RAR) **archives** for **gallery** view is too slow, you may want to disable it. This will also disable search hit context preview for search hits in files in archives.

- If large JPEGs already contain embedded thumbnails and those have been included already in the volume snapshot or if internal thumbnails have been computed for large pictures, then they can be optionally used as **auxiliary thumbnails** in the **gallery** to represent the main picture. The benefit is that they are of course *much* quicker to load than the main large picture. Also video stills exported from videos can be used as auxiliary thumbnails to represent the video, even all of them dynamically rotating if fully checked.

- The gallery has its own "Dbl-click=View instead of Explore" 3-state option, analogously to the directory browser. By default, double-clicking means View in the gallery.

- There is an option to view files with a single click in the gallery instead of with a double click. Useful for example if you wish to view certain pictures on a separate monitor, where you do not have to close the view window to see the gallery again, when not viewing all pictures one after the other (for which the Page Up or Dn key is more efficient).

- Another option allows to tag a file by clicking anywhere in the thumbnail, not just in the tag square. That makes it more convenient to tag a large number of files, and is more comfortable that selecting multiple files while holding the Ctrl key.

- The gallery can optionally show thumbnails for any file type supported by the viewer component, including Office documents, PDF, HTML, e-mails, and pictures that the internal graphics viewing library cannot display (e.g. .emf, .wmf, ...). You can choose between normal and slightly shrunk and strongly shrunk thumbnails of documents. Shrunk thumbnails show much more detail from an original document and the original layout, but at the cost of readability. Larger fonts (in particular captions) in an original document, if not shrunk, are typically readable in the thumbnail and can already give you an idea what kind of document it is even if don't view it, so you can more quickly find the documents that you are looking for.

Plus, you will be able to see which files can be nicely viewed with the viewer component at all. It is strongly recommended run X-Ways Forensics with Aero enabled in Windows when using the gallery with the non-picture option.

Files that are larger than 16 MB are not represented with a thumbnail, for performance reasons. X-Ways Forensics tries to abort the generation of a thumbnail if it takes longer than a few seconds. If the generation of a true thumbnail is unsuccessful, you may see a viewer component error message like "Operation cancelled" in tiny red letters in the thumbnail instead. If thumbnail generation is not even attempted by X-Ways Forensics, you will just see the filename and an icon.

- You may specify your **preferred thumbnail size** in pixels. WinHex will decrease the size automatically if needed to ensure that at least as many files are displayed in the gallery view as are displayed in the currently visible section of the directory browser.

- The timeout in milliseconds when loading pictures with the internal graphics viewing library is aborted (e.g. corrupt or unsupported or extremely large picture files), is user-definable.

Keeping Track of Viewed Files

With a forensic license, the program can optionally keep track of which files were already viewed and flag them visually with a green background color around the tag. This is especially useful when reviewing hundreds or thousands of documents or pictures over a longer period, to avoid accidentially viewing the same documents multiple times. A file can automatically be flagged as already viewed when viewing it in full window or Preview mode, when viewing pictures in the gallery, or when identifying a file as known good based on the hash database.

When identifying duplicate files based on hash values, and one of the files has been marked as already viewed, then the duplicates can optionally be marked as already viewed, too. Similarly (only if the corresponding checkbox is fully checked), if files have been marked already as having duplicates and their hash values are available, when they are viewed, known duplicates within any open volume will be marked as already viewed at the same time, but this is potentially slow when used in conjunction with the gallery. When viewing a file with further hard links (which are also duplicates), those will be automatically marked as already viewed as well, except in HFS+.

To manually mark files as already viewed, you can press Alt in combination with the cursor keys. Alt+Left removes the mark. You can also right-click the tag area of a file in the directory browser to mark it as already viewed or to remove that mark.

A directory is considered viewed if all the files and subdirectories that it contains are flagged as such.

9.4 Undo Options

The availability of the “Undo” command depends on the following options:

- Specify how many sequential actions are to be reversed by the Undo command. This option does not affect the number of reversible keyboard inputs, which is only limited by the available RAM.

- In order to save time and space on your hard disk, you can specify a file size limit. If a file is larger than this limit, backups will not be created and the Undo command is not available except for keyboard input.

- Automatically created backups for the internal use with the Undo command are deleted by WinHex when closing the file, if the corresponding option is fully selected. If it is partially selected, they are deleted when WinHex terminates.

- For all kinds of editing operations you choose whether they should be reversible or not. If so, an internal backup is created before the operation takes place.

9.5 Security Options

Use the option **Check for virtual memory** alteration to make sure the memory editor inspects the structure of virtual memory every time before *reading* from or *writing* to it. If the structure has changed, a possible read error is prevented. Especially under Windows NT the checking may result in a loss of speed. When editing the "entire memory" of a process, WinHex generally *never* checks for alterations before reading, even if this option is enabled.

Before modifications to an existing file are saved (i.e. before the **file** is **updated**), you are prompted for **confirmation**. To inhibit this behavior of WinHex, switch off the corresponding option.

If any of the operations Refine Volume Snapshot, Logical Search, or Indexing crashes when processing a file, X-Ways Forensics when started next time will tell, which file was likely responsible for the crash, if you had it **collect information for** a **crash report**.

Output messages about exceptions: Determines the verbosity of the program in case of exception errors. If totally unchecked, only exception errors with a potentially serious impact (like considerably incomplete analysis results) will be brought to your attention in the Messages window. If fully checked, all of them will be output, even those that occur typically with corrupt files only and have no negative impact on other analysis results. The middle state is a reasonable compromise. Regardless of this option, exception errors will be noted in the error.log file.

All notices and warnings output to the **Messages** window can optionally be automatically saved in a text file "**msglog.txt**" in the installation directory. If at that time a case is active, the notice/warning will be written to the msglog.txt file in the log subdirectory of that case instead.

Strict drive letter protection: Only available with a forensic license.Active by default in X-Ways Forensics. Ensures that saving and editing files is only possible on certain drive letters, namely those that X-Ways Forensics even when examining a live system can assume are located

on the examiner's own media. They are: 1) the drive letter that hosts the active case if one is active, 2) the drive letter with the directory for temporary files, 3) the drive letter from which X-Ways Forensics was run and 4) the drive letter that contains the directory for image files.

The **key** that is required for encryption and decryption can be entered in a normal edit box. Optionally, you **enter** it **blindly** (asterisks are displayed instead of the actual characters). In this case you have to confirm the key in a second edit box to detect typos.

By default, the **key** is **kept in** main **memory** (in an encrypted state) as long as WinHex is running, so that you do not have to type it again and again if you use it several times. Possibly you prefer WinHex to erase the key after use.

Decide whether or not WinHex shall **prompt before executing a script**, or only before executing a script via the command line.

9.6 Search Options

Case sensitive: If a search is case-sensitive, that means that upper and lower case characters are distinguished and e. g. “Option” with a capital “O” is not found in the word “optionally”. By unchecking the checkbox, you search for all upper-case/lower-case variants of the search terms. Searches are fully case insensitive only with the Simultaneous Search, with the Find Text command only for letters from the Latin/English alphabet and German umlauts. In the Simultaneous Search you may use case-sensitive and non-case-sensitive search terms at the same time if the “Match case” option is half selected. In that case you may prepend search terms with “case:” to mark them as case-sensitive.

Unicode: The specified text is searched in UTF-16 Little Endian. The simultaneous search allows to search for the same text at the same time in Unicode and in other code pages.

You may specify a **wildcard** (one character or a two-digit hex value), which represents one byte. For example this option can be used to find “Speck” as well as “Spock” when searching for “Sp?ck” with the question mark as the wildcard.

Only whole words: The search term is found only if it occurs as a whole word, i.e. if delimited from other words by any character other than a...z, A..Z and German and French letters (e. g. by punctuation marks, blanks, binary control codes, digits). If this option is enabled, for example "tomato" is not found in "automaton". Reliable to reduce the number of hits for English, German, and French text only. In a Simultaneous Search either all search terms are searched as whole words or only those that are indented (prepended with a tab character) or none, depending on the state of the corresponding check box. If you wish to combine the indention for a search as a whole word with the "case:" prefix for case sensitivity, enter the "case:" prefix first and then insert the tab character for the indention.

For a Simultaneous Search you may customize the word boundary detection for Latin-based languages, i.e. make it more strict (for less search hits) or more relaxed (for more search hits), by defining the alphabet of characters that are considered letters (characters belonging to words) as

opposed to non-word characters. A word character followed by a non-word character or the other way around is considered a word boundary. There are three easy-to-use pre-defined settings. The setting for the most thorough search results is the default. Users that are overwhelmed by garbage hits for short keywords in non-text data such as Base64 or binary garbage may want to try the other two options. These other two options could lead to valid search hits being missed in some constellations (depends on the file format), but can still be justifiable as a great time saver for searches in text documents, e.g. rather in electronic discovery, rather not in computer forensics.

For more explanation and an example of how the whole words option works, please read on: A word boundary is a boundary between 2 consecutive characters of which one character is a word character and the other character is not a word character. If both characters are word characters (e.g. "ns"), then obviously the "s" does not start a new whole word, and the "n" cannot be the end of a whole word. It can be somewhere in the middle of a whole word (e.g. "mansion"), but in between these two characters "ns" there is definitely no word boundary. If both characters are non-word characters (e.g. "! ", exclamation mark followed by a space), then obviously the position between the two is not a word boundary either. The exclamation mark cannot be the end of a word (cannot occur anywhere within a word), and the space cannot be the start of a word (cannot occur anywhere within a word either, excluding compound words).

If you are searching for "man" as a whole word within "our mansion", then XWF will provisionally/internally find "man", and then first check whether the character before the "m" is a word character. That character is a space. A space character is not a word character. Then it also checks whether "m" is a word character according to the alphabet. It is. That means there is a word boundary before the "m". Next XWF needs to check whether "n" and "s" are word characters. Both are. That means that after the "n" there is no word boundary. Hence the three letters "man" within "mansion" are not considered a whole word occurrence of "man".

Search direction: Decide whether WinHex shall search from the beginning to the end, or downwards or upwards from the current position.

Condition: Offset modulo x = y: The search algorithm accepts search string occurrences only at offsets that meet the given requirements. E.g. if you search for data that typically occurs at the 10^{th} byte of a hard disk sector, you may specify x=512, y=10. If you are looking for DWORD-aligned data, you may use x=4, y=0 to narrow down the number of hits.

Search in block only: The search operation is limited to the current block.

Search in all open windows: The search operation is applied to all open edit windows. Press F4 to continue the search in the next window. If “Search in block only” is enabled at the same time, the search operation is limited to the current block in each window.

Count occurrences/Save occurrence positions: Forces WinHex not to show each single occurrence, but to count them. If this option is fully enabled, WinHex will enter all occurrences into the Position Manager.

Search for “non-matches”: In “Find Hex Values” you may specify a single hex value with an exclamation mark as a prefix (e.g. !00) to make WinHex stop when it encounters the first byte

value that *differs*.

GREP syntax: Search option available with the Simultaneous Search only. Regular expressions are a powerful search tool. A single regular expression may match many different words. Either all search terms are considered GREP expressions or only those prepended with "grep:" or none, depending on the state of the corresponding checkbox. You may prepend a search term with both "case:" (see above) and "grep:" in that order. The following characters have a special meaning in regular expressions, as explained below: ( ) [ ] { } | \ . # + ?. Where these special characters are to be taken literally, you need to prefix them with a backslash character (\).

The | operator is used to denote alternative matches. You can use the regular expression *car (wheel|tire)* to search for the words "car wheel" and "car tire". Any match must equal the parts before, after, or between any | operators present. The effect of | is only limited by parentheses.

. and # are wildcards: . matches any character, # matches any numeric character. You can define sets of characters with the help of square brackets: [xyz] will match any of the characters x, y, z. [^xyz] will match any character except x, y, or z. You can define ranges of characters using a dash: [a-z] matches any lower-case letter. [^a-z] matches all characters except lower-case letters. The listing may comprise individually listed characters and ranges at the same time: [aceg-loq] matches a, c, e, g, h, i, j, k, l, o, and q. All characters except [,], -, and \ are taken literally between square brackets, even the wildcard characters . and #.

\b stands for the start or end of a word, i.e. the boundary between a word character and a non-word character. Which characters/letters are considered word characters by the Simultaneous Search is user-defined. The start and end of a file also count as word boundaries. \b is only supported at the start and/or at the end of the search term, and not in conjunction with |. \b, ^, and $ anchors only work only when searching in evidence objects of a case, and not for index searches.

Byte values that correspond to ASCII characters that cannot be easily produced with a keyboard can be specified in decimal or hexadecimal notation: For example, \032 and \x20 are both equivalent to the space character in the ASCII character set. This kind of notation is supported even in between square brackets. E.g. [\000-\x1f] matches non-printable ASCII characters.

Multiplier characters (*, +, and ?) indicate that the preceding character(s) may or must occur more than once (see below). Complex example: a(b|cd|e[f-h]i)*j matches aj, abj, acdj, aefij, aegij, aehij, abcdj, and abefij.

Within [] brackets, the characters .*+?{}()| are not treated as special characters, but literally.

Brief overview of supported syntax features (everything else is interpreted literally)

.	A period matches any single character.
#	A pound sign matches any numeric character [0-9].
\nnn	A byte value specified with three decimal digits (0...255)
\xnn	A byte value specified with two hexadecimal digits (0…FF). For example, \x0D\x0A is the Windows line break.
\unnnn	A Unicode value specified with four hexadecimal digits. Depending on the selected code page(s), corresponds to different byte values.

? Matches one or zero occurrences of the preceding character or set.
* Matches any number of occurrences of the preceding character, including zero time.
+ A plus sign after a character matches any number of occurrences of it except zero.
[XYZ] Characters in brackets match any one character that appears in the brackets.
[^XYZ] A circumflex at the start of the string in brackets means NOT. For 8-bit searches only.
[A-Z] A dash within the brackets signifies a range of characters.
\ Indicates that the following special GREP character is to be treated literally.
{X,Y} Repeats the preceding character or group of characters X-Y times.
(ab) Functions like a parenthesis in a mathematical expression.
Groups ab together for +, ?, *, | and {}.
a | b The pipe acts as a logical OR. So it would read "a or b".
\b Matches a word boundary.
^ Matches the start of a file.
$ Matches the logical or physical end of a file, depending on the search options.

GREP Examples

E-mail addresses
[a-zA-Z0-9_\-\+\.]{1,20}@[a-zA-Z0-9\-\.]{2,20}\.[a-zA-Z]{2,7}
(the + before the @ is supported in Gmail addresses)

Internet addresses starting with http://, https://, ftp://
[a-zA-Z]+://[a-zA-Z0-9/_\?$&=\-\.]+

Visa and Mastercard credit card numbers
[^#a-z][45]###############[^#a-z]
[45]###-####-####-####
[45]### #### #### ####
(ideally check results via an X-Tension with the Luhn algorithm to reduce the number of false hits and search without [^#a-z])

Allow overlapping hits: If you use GREP syntax to search for search hits of variable length, multiple valid hits at the same location may be the result. If you search for example for e-mail addresses, and the search algorithm is fed with the character sequence "mail@x-ways.com", then it will determine that the characters from the "m" in "mail" match the GREP expression and it will record a hit. After that, it proceeds with the "a" in "mail" and realizes, that ail@x-ways.com fits the bill as well, and so do il@x-ways.com and l@x-ways.com. All of these might be valid e-mail addresses. So the search algorithm is entirely right, but typically users do not wish to see those additional hits. So if you do not allow for overlapping hits, new hits are recorded only after the "m" in ".com". Not allowing overlapping hits means to exclusively assign the characters covered by a hit to that hit and not to potential other hits any more.

Search window, proximity searches

The GREP search window width is 128 bytes by default. That means it is not guaranteed that with a variable-length GREP search term (i.e. using {}*+ syntax) you can find data that is longer than 128 bytes. You may increase the search window width if you need to cover more than that.

This is needed for example for proximity searches. If you require that a document contains two search terms at the same time, and that the search terms should occur close to one another, you could search for these search terms with two GREP expressions and specify the maximum distance allowed between them as the second parameter in the braces:

keyword1.{0,*maxdistance*}*keyword2*

keyword2.{0,*maxdistance*}*keyword1*

The search window width in bytes required when searching with an 8-bit character set is the sum of *maxdistance*, length(*keyword1*) and length(*keyword2*).

Please note that the preferred method to find two search terms near to each other is the NEAR combination in the search term list, when two search terms are already combined with a logical AND, after they have been searched for separately.

9.7 Replace Options

Prompt when found: WinHex awaits your decision when an occurrence has been found. You may either replace it, continue or abort the search.

Replace all occurrences: All occurrences are replaced automatically.

Case sensitive: The characters that are to be replaced are searched using this option (cf. Search Options).

Unicode character set: The specified characters are searched and replaced in Unicode format (cf. Search Options).

You may specify one character or a two-digit hex value as a **wildcard**. This is usually done in the search string. If the *substitute* contains a wildcard, the character at the corresponding position in an occurrence will not be changed. Thus, “black” and “block” can be replaced simultaneously with “crack” and “crock” (enter “bl?ck” and “cr?ck”).

Only whole words: The searched string is recognized only if it is separated from other words e.g. by punctuation marks or blanks. If this option is enabled, “tomato” is not replaced in “automaton”.

Search direction: Decide whether WinHex shall replace from the beginning to the end, or downwards or upwards from the current position.

Replace in block only: The replace operation is limited to the current block.

Replace in all opened files: The replace operation is applied to all files not opened in view mode. If “Replace in block only” is enabled at the same time, the replace operation is limited to the current block of each file.

Hint:

WinHex is able to replace one string or hex value sequence with another one that has a different length. You will be prompted, which of the following methods shall be applied:

1st method: The data behind the occurrence is moved due to length difference. So the file size is changed. This method must not be applied to certain file types, such as executable files. It is even possible to specify nothing as the substitute, which means all occurrences will be removed from the file!

2nd method: The substitute is written into the file at the position of the occurrence. If the substitute is shorter than the searched character sequence, the exceeding characters will remain in the file. Otherwise even the bytes behind the occurrence will be overwritten (as far as the end of the file is not reached). The file size is not affected.

10 Miscellaneous

10.1 Block

You can mark a range of bytes or sectors of an open file or disk as a “block”. This part can be manipulated by several function in the edit menu just as selections in other Windows programs. If no block is defined, these functions usually are applied to the whole file or disk.

The current position and size of the block are displayed in the status bar. Double-clicking the right mouse button or pressing the **ESC** key clears the block.

10.2 Modify Data

Use this command to modify the data within the block or within the whole file, in case no block is defined. In this version of WinHex, four types of data modifications are available. Either a fixed integer number is added to each element of the data, the bits are inverted, a constant is XORed with the data (a simple kind of encryption), ORed, or ANDed, bits rotated left in a circular pattern (first byte rotated by 1 bit, second byte by 2 bits, and so on), bits are shifted logically, or bytes are swapped. By shifting bits, you can simulate inserting or removing single bits at the beginning of the block. You may also shift entire *bytes* (currently to the left only, by entering a negative number of bytes). This is useful if you wish to cut bytes from a very huge file in in-place mode, which would otherwise require the creation of a huge temporary file.

Swap Bytes

This command assumes all data to consist of 16-bit elements (32-bit elements resp.) and swaps high-order and low-order bytes (and high-order and low-order words resp.). Use it in order to convert big-endian into little-endian data and vice versa.

Addition

Specify a positive or negative, decimal or hexadecimal number, which is to be added to each element of the current block. An integer format defines size (1, 2 or 4 bytes) and type (signed or unsigned) of an element.

There are two ways how to proceed if the result of the addition is out of the range of the selected integer format. Either the range limit is assumed to be the new value (I) or the carry is ignored (II).

Example: unsigned 8-bit format
I. FF + 1 → FF (255 + 1 → 255)
II. FF + 1 → 00 (255 + 1 → 0)

Example: signed 8-bit format
I. 80 - 1 → 80 (-128 - 1 → -128)
II. 80 - 1 → 7F (-128 - 1 → +127)

- If you decide to use the first method, WinHex will tell you, how often the range limit has been exceeded.

- The second method makes sure the operation is reversible. Simply add -x instead of x based on the same integer format to recreate the original data.

- When using the second method it does not make a difference whether you choose a signed or an unsigned format.

10.3 Conversions

WinHex provides the Convert command of the Edit menu for easy conversions of different data formats and for encryption and decryption. The conversion can optionally be applied to all opened files instead of only the currently displayed one. The formats marked with an asterisk (*) can only be converted as a whole file, not as a block. The following formats are supported:

- ANSI ASCII, IBM ASCII (two different ASCII character sets)
- EBCDIC (an IBM mainframe character set)
- Lowercase/uppercase characters (ANSI ASCII)
- Binary* (raw data)
- Hex ASCII* (hexadecimal representation of raw data as ASCII text)
- Intel Hex* (=Extended Intellec; hex ASCII data in a special format, incl. checksums etc.)
- Motorola S* (=Extended Exorcisor; ditto)
- Base64*
- UUCode*
- Percentage URL Encode
- Quoted Printable

Please note:

- When converting Intel Hex or Motorola S data, the internal checksums of these formats are not checked.
- Depending on the file size, the smallest possible output subformat is chosen automatically. Intel Hex: 20-bit or 32-bit. Motorola S: S1, S2, or S3.
- When converting from binary to Intel Hex or Motorola S, only memory regions not filled with hexadecimal FFs are translated, to keep the resulting file compact.

The Convert command can also decompress any number of complete 16-cluster compression units compressed by the NTFS file system* and (with a forensic license) entire hiberfil.sys files that were copied off an image as well as individual xpress chunks from such files. Also, it allows to convert so-called Nandroid backup files of the NAND flash memory of Android devices to regular raw images.

Furthermore it can stretch packed 7-bit ASCII to readable 8-bit ASCII*, useful e.g. for SMS from mobile phones.

Encryption/Decryption

Specify a string consisting of 1-16 characters as the encryption/decryption key. The key is case-sensitive. The more characters you enter, the safer is the encryption. The key itself is not used for encryption and decryption, instead it is digested to the actual key. The key is not saved on your hard disk. If the corresponding security option is enabled, the key is stored in an encrypted state in the RAM as long as WinHex is running.

It is recommended to specify a combination of at least 8 characters as the encryption key. Do not use words of any language, it is better to choose a random combination of letters, punctuation marks, and digits. Note that encryption keys are case sensitive. Remember that you will be unable to retrieve the encrypted data without the appropriate key. The decryption key you enter is not verified before decrypting.

Encryption algorithm: 256-bit AES/Rijndael, in counter (CTR) mode. This encryption algorithm uses a 256-bit key that is digested with SHA-256 from the 512-bit concatenation of the SHA-256 of the key you specify and 256 bits of cryptographically sound random input (“salt”). The file is expanded by 48 bytes to accommodate the 256 bits of salt, and a randomized 128-bit initial counter.
WinHex allows you to encrypt not only an entire file, but also a block of data only. In that case you are warned, however, that no salt is used and no random initial counter is used, so you must not reuse your key to encrypt other data with the same encryption method. The size of the block is left unchanged.

10.4 Sector Superimposition

With this feature you can superimpose other data on top of disks or interpreted images that are opened as read-only. Useful when you need to make minor temporary adjustments to data in

sectors within the program to get it interpreted correctly internally, but do not want to or are not allowed to alter the sectors on the disk or in the image itself (or cannot because it is not a raw image, but an .e01 evidence file) and also do not want to make another complete working copy of an image that is e.g. 2 TB in size if just 1 byte needs to be changed. Such adjustments can be necessary for example in cases of partitioning or file system metadata corruption, where just a missing magic number keeps WinHex from detecting the file system or just one flipped bit keeps WinHex from finding $MFT in NTFS or just one wrong nibble in the partition table keeps WinHex from recognizing a partition as an LVM2 container partition etc. etc. In these situations you can manually provide and superimpose the corrected data and then hopefully work with the disk or image with no further problems, getting all partitions and files listed immediately as if nothing was wrong. This functionality is intended for advanced users that do not give up easily when at first they see "nothing" and have some understanding of low level data structures and know how to fix them.

You can enable and disable superimposition for the disk or partition in the active data window using the Edit | Superimpose Sectors menu command. This command allows you to select any file with the raw contents of disk sectors. For example, you can create such a file by selecting one or more sectors as a block, copying the block into a new file, making the necessary adjustments (possible even in X-Ways Forensics because ordinary files unlike disks or interpreted images can be edited) and saving that file. When applied, the contents of this file are superimposed to the sectors starting with the sector in which the cursor is located, or if the file is named "*n*.sector", where *n* is a number, it will be applied to the sectors starting with sector *n*, and all other files in the same directory matching the same mask will also be applied to sector numbers as indicated within the filename. You will immediately see the superimposed data when navigating to the affected sectors, and can continue making adjustments to the imposed raw data file if you keep it open in a separate window. As soon as you have saved changes in that window, they will take effect in the data window that represents the disk or partition whose data you are trying to fix when you refresh the view, take a new volume snapshot, define the start of a partition, try again to open a file with a corrupt FILE record etc. etc.

Please note that only complete sectors, not partial sectors, can be superimposed. Superimposition can be active only for one disk or disk partition or image at a time. If desired, you can make a copy (image or cloned disk) of the virtually repaired disk or image with the usual commands while the superimposition is in effect, so that the copy will have the superimposed sectors directly embedded.

10.5 Wiping and Initializing

To securely erase (shred) data in disk sectors, unused disk areas (Disk Tools menu), or files selected with the Wipe Securely command, and also simply to fill files with certain byte values, WinHex offers the following options:

With constant byte values specified in hexadecimal notation: Specify either 1, 2, 3, 4, 5, 6, 12, 15, or 16 two-character hex values, which will be copied repeatedly into the current block, the entire file or all disk sectors, respectively. Very fast.

With simple pseudo-random byte values: Specify a decimal interval (0 to 255 at max.) for random numbers, which will be copied repeatedly into the current block, the entire file or all disk sectors, respectively. The random bytes are Laplace-distributed. Fast.

With pseudo-random data that simulates encryption: Random data that is supposed to be indistinguishable from encrypted data. Quite fast.

With cryptographically sound pseudo-random data: Cryptographically secure pseudo-random number generator (CSPRNG) named ISAAC, *very* slow.

In case in all open files *either a block or no block is defined*, this command can optionally be applied to all these files at the same time.

To maximize security, if you wish to totally wipe (sanitize) slack space, free space, unused NTFS records, or an entire media, you may want to apply more than one pass for overwriting disk space (up to three).

According to the Clearing and Sanitization Matrix, the standard outlined in the U.S. Department of Defense (DoD) 5220.22-M operating manual, method "c", a hard disk or floppy disk can be cleared by overwriting (once) all addressable locations with a single character. This is usually the hexadecimal value 0x00, but can be any other value. To sanitize hard disks according to method "d", overwrite all addressable locations with a character, its complement, then a random character, and verify. (This method is not approved by the DoD for sanitizing media that contain top secret information.)

The "DoD" button configures WinHex for sanitization, such that it will first overwrite with 0x55 (binary 01010101), then with its complement (0xAA = 10101010), and finally with random byte values.

The "0x00" button configures WinHex for simple initialization, wiping once with zero bytes.

10.6 Disk Cloning

Tools | Disk Tools | Clone Disk. This function copies a defined number of sectors from a source to a destination. Both the source and the destination can be either a *disk* (click the button with the disk icon) or a *file* (click the button with the file icon).

In case both the source and the destination are disks, both disks must have the same sector size. In order to effectively *duplicate* a medium (i.e. copy all sectors), simply copy *all* sectors. Select the appropriate option, so the correct number of sectors is entered automatically. The destination disk must not be smaller than the source disk. As a *disk* you can also select an interpreted image or a partition opened from within a physical disk in the background. As a target you cannot select an interpreted .e01 evidence file as such images cannot be rewritten, only raw images. As a *file* you can only specify unsegmented raw images, e.g. .dd, .001, .img etc., no other image types such as .e01, .vhd., .vmdk etc.

Disk cloning offers options that control the behavior when bad sectors are encountered on the source disk:

- By default, you are notified of the error and prompted for either continuing or aborting the operation. “Log procedure silently” creates a complete log file of the entire operation in the folder for temporary files (filename “Cloning Log.txt”), including a report on unreadable sectors (which cannot be copied), and prevents WinHex from reporting each unreadable sector separately.
- WinHex can either leave a destination sector that corresponds to a damaged source sector unchanged or fill it with an ASCII pattern you specify (e.g. your initials, or something like “BAD ”). Leave the pattern edit box blank to fill such sectors with *zero* bytes. BTW, the chosen pattern is also used to display a bad sector's contents in the disk editor.
- Bad sectors often occur in contiguous groups, and each attempt to read a bad sector usually takes a long time. You may have WinHex avoid such damaged disk areas. When a bad sector is encountered, WinHex can skip a number of subsequent sectors you specify (32 by default). This is useful if you wish to accelerate the cloning process and if you do not care about some actually readable sectors not making it to the clone.

Regular disk cloning is not an option if you want to duplicate a disk in a removable drive (e.g. a floppy disk) with only one removable drive present. The correct concept for this application is *disk imaging*, where the data is first stored in an image *file*. The image can then be copied to a different disk. The result is the same as disk cloning.

When you specify a file named “dev-null” as the destination, the data will only be read and not copied anywhere (and you will be warned of this). This is useful if you are interested in the report about bad sectors, but do not wish to actually clone or image a disk.

You may try “simultaneous I/O” if the destination is not the same physical medium as the source. Offers a chance to accelerate the cloning process by up to 30%.

Specialist license or higher: In conjunction with simultaneous I/O you may also have WinHex copy the sectors of a disk in *reverse* direction, *backwards* from the end of the source disk. Useful if the source disk has severe physical defects that for example cause a disk imaging program or your entire computer to freeze or crash when reaching a certain sector. In such a case you can additionally create an image in reverse order, by reading sectors from the disk backwards one by one, or better, you can even automatically *complete* an existing incomplete unsegmented conventional ("forward") raw image from the rear end to get an image that is as complete as possible, filled from both ends, with ideally only a small zeroed gap in the middle that represents the unreadable damaged spot on the source hard disk. For that you simply select an incomplete raw image file that you already have as a destination file, and you will be asked whether you wish to complete it instead of overwrite. WinHex will do the rest, e.g. allocate the missing sectors in the image file (zeroed out) so that it has the complete size of the source disk and then fill the file backwards as much as possible. Be sure to create reverse images on NTFS volumes, not FAT32. The source start sector to specify for reverse imaging is the same as for conventional forward images, i.e. usually 0 when imaging a complete hard disk.

For disk imaging in general it is recommended to use the File | Create Disk Image functionality instead, for various reasons (with a forensic license: support for .e01 evidence files, compression,

splitting, hashing, encryption, metadata, technical details report, more convenient). Only in specific cases, for example when dealing with several physical disk defects or when the goal is to copy only certain ranges of sectors, advanced users can use Tools | Disk Tools | Clone Disk to have more detailed control over which sectors are copied from where to where in which order.

10.7 Images and Backups

This command "Create Disk Image"/"Make Backup Copy" in the File menu allows to create a backup or image of the currently open logical drive, physical disk, or individual file. There are three possible output file formats, each with unique advantages.

File format:	WinHex Backup	**Evidence File**	**Raw Image**
Filename extension:	.whx	.e01	e.g. .dd
Interpretable as disk:	no	yes	yes
Splittable:	yes	yes	yes
Compressible:	yes	yes	no
Encryptable:	no	yes	no
Optional hash:	integrated	integrated	separate
Optional description:	integrated	integrated	separate
Range of sectors only:	yes	(yes)	(yes)
Applicable to files:	yes	no	no
Automated maintenance:	Backup Manager	no	no
Compatibility:	no	(yes)	yes
Required license:	none	forensic	personal

The major advantage of evidence files and raw images is that they can be interpreted by WinHex like the original disks (with the command in the Specialist menu). This also makes them suitable for usage as evidence objects in your cases. This holds true for evidence files in particular because they can store an optional description and an integrated hash for later automated verification. Raw images have the benefit that they can be easily exchanged between even more forensic tools. All output file formats support splitting into segments of a user-defined size. A segment size of 650 or 700 MB e.g. is suitable for archiving on CD-R. Evidence files must be split at 2047 MB at most to make them compatible with X-Ways Forensics versions before v14.9 and EnCase versions before v6 and certain other tools. With a forensic license, raw image files and evidence files can automatically be verified immediately after creation, by recomputing the hash value that was originally computed from the medium, with the image instead.

Evidence file and WinHex backup compression is based on the "Deflate" compression algorithm that is part of the popular general-purpose library *zlib*. This algorithm consists of LZ77 compression and Huffman coding. With the "normal" compression level you can reach a compression ratio of 40-50% on average data. However, this comes at the cost of a considerably reduced imaging speed. "Fast/adaptive" compression is a *very good* and *intelligent* compromise between speed and good compression, not like the ordinary fast compression option in other programs. With "high" compression you gain only a few percentage points more compression, but at disproportional high cost. For WinHex backups, "adaptive" is the same as "normal".

Raw image files can be compressed at the NTFS file system level, if they are created on NTFS volumes. Either normal NTFS compression is used, or the image file can be made "sparse", such

that large amounts of zero-value bytes won't need drive space.

Cleansed images: With a forensic license, there is an acquisition option for those users who need to or want to exclude certain files from forensic images, called "Omit excluded files". The data stored in clusters that are associated with files that you exclude before starting the imaging process can automatically be zeroed out in the image. That won't have any effect on files whose contents are not stored in their own clusters. Before you start the imaging process for a partitioned disk, open the partitions in which the files are located that you would like to exclude. Wait till the volume snapshot has been taken if it was not taken before. Then exclude the files. You do not need to open and take volume snapshots of partitions whose data you would like to include completely. All other data is copied to the image normally. There is an option to "watermark" wiped sectors in the image with an ASCII or Unicode text string, so that when working with the image you are reminded of the omission when you look at the affected areas. Cleansed images are useful for anyone who needs to redact certain files in the file system, but otherwise wants to create an ordinary forensically sound sector-wise image, compatible with other tools. A must in countries whose legislation specially protects the most private personal data of individuals and certain data acquired from custodians of professional secrets (e.g. lawyers and physicians, whose profession swears them to secrecy/confidentiality). Limitation: Not available for disks partitioned as Windows dynamic disks or with Linux LVM*. Only files in supported file systems can be omitted. Note that you can also retroactively cleanse (redact) already created conventional raw images, in WinHex, by securely wiping files selected files via the directory browser context menu. The granularity of this operation is not limited to entire clusters. For example, that means it can also wipe files in NTFS file systems with so-called resident/inline storage and it does not erase file slack along. For a comparison of evidence file containers, skeleton images and cleansed images please see our web site. All of those are images that only transport a subset of the original data.

Another kind of cleansed image is an image in which all the clusters marked by the file system as free are zeroed out (specialist or forensic license only). That is very useful if you create the image for backup purposes and not for forensic purposes, or if for forensic purposes you do not require data in free space or are not supposed to acquire it (to only examine existing files). In conjunction with compression, this option has the potential to save a lot of drive space, depending on how much free space there is, and imaging speed can be greatly accelerated if there are large contiguous free drive space areas in volumes/partitions. Note that in case of file system inconsistencies clusters could be erroneously regarded as free. You have to specifically confirm the creation of cleansed images as in the traditional sense they are not forensically sound (though in a more modern sense of the word they can be, depending on the jurisdiction that you work in in countries with stricter personal privacy rights and depending on the overall situation).

X-Ways Forensics checks for and warns of overlapping partitions when creating a cleansed image of a partitioned physical disk. Clusters in affected disk areas are not omitted. In such a situation, it is recommended to image the relevant partitions separately.

Forensic license: When creating an image, the technical details report is created and written to a text file that accompanies the image file. For an .e01 evidence file it is also incorporated directly into the .e01 file as a description. The SMART information is queried and written to the text file again upon completion of the image, so that you can see whether the status of a hard disk in bad shape has further deteriorated during imaging. Secondly, you can see how the "power on time"

has changed, which is useful to deduce its unit of measurement (usually hours, but can be different on certain hard disk models). The text file also indicates the amount of time spent creating the image, the compression ratio achieved, the result of an immediate verification of the image based on the hash value (if selected), and any sector read errors.

Forensic license: Ability to create a second copy of an image immediately when imaging a disk, which is much quicker than copying the image file later and makes sense if the 2nd copy is created on a different drive. File spanning (i.e. when to start another image file segment) is kept in sync between both copies even when running out of space on one of the two target drives only.

Forensic license: Ability to compute two hash values simultaneously. If you make use of this option, then both hash values will be stored in the descriptive text file. The first hash value is the one that can be automatically verified when imaging completes. You could intentionally choose the faster algorithm for that as the main purpose at that point is to detect I/O errors and file errors. The second hash value is imported into the evidence object properties when adding the image to a case.

Forensic license: Ability to schedule in advance subsequent disk imaging operations in additional instances that will wait until already ongoing imaging operations in previous instances have completed, to avoid inefficient simultaneous creation of multiple images on the same output disk (which is unnecessarily slow and produces highly fragmented image files). Additional instances only wait for previous instances in which the checkbox for waiting was checked as well, but not for others.

Forensic license: If you cancel disk imaging in the middle of the process, X-Ways Forensics quickly finalizes the .e01 evidence file format (more precisely, the current segment) to guarantee a consistent image even though it is not a complete image. Useful for example in an emergency situation when imaging media on site, because a incomplete image that can be used without errors is better than an unusable corrupt image. If hashing was enabled, incomplete .e01 images even have a hash value that can later be verified later.

Forensic license: For the .e01 evidence file format, you may choose the internal chunk size. Might be regarded as useful by some to achieve a marginally better compression ratio for ordinary data, at the expense of more time needed when creating the image and when later randomly accessing data in the image, but improves compression noticeably for extremely compressible data (e.g. a wiped or unused areas of a hard disk). A 512 KB chunk size reduces the image size with ideal data (e.g. only 0x00 bytes) ceteris paribus by an additional 40% compared to a 32 KB chunk size.

Forensic license: You may adjust the compression option while .e01 evidence files are being created. Useful if your priorities (higher compression rate or higher speed) change, for example when you see that drive space suddenly seems scarce or you have to finish the process quicker than previously thought. Also useful to experiment, when not sure which compression option might be best for a particular system configuration (e.g. when imaging a live system on site and having to write the image to an external hard disk via USB, where I/O is slow and the overall process may be faster with compression than without).

Forensic license: When imaging with active compression in .e01 format, X-Ways Forensics

provides immediate visual feedback about the actual amount of data found on the disk. That is possible because disk areas that were never written as well as disk areas that were wiped achieve extremely high compression ratios. The rolling compression ratio is represented during imaging by vertical bars in a separate window. The higher the bar, the lower the "data density" in that area. The compression statistics are also stored in the .e01 evidence file, so that the same chart is also available at any later time from the evidence object properties dialog when you click the "Compression" button.

Forensic license: Ability to specify how many extra threads to use for compression when creating .e01 evidence files. By default X-Ways Forensics will use no more than 4 or 8, and it depends on how many processor cores your system has, but you could try to increase the number on very powerful systems with even more cores usually without problems, for a chance to further increase the speed, or you can reduce it you run into stability problems.

Forensic license: You have the option to change the nature of an image (disk or volume) and its sector size when creating the image. This is possible not only for .e01 evidence files, where both is explicitly defined in the internal metadata (compatible with other tools), but also for raw images (via external metadata, compatible only with X-Ways Forensics/Image v18.4 and later, lost if the image leaves the realm of NTFS file systems). Useful whenever the source of the data is not an ideal interpretation. For example if a reconstructed RAID actually represents a volume, not a physical disk, then you can already adjust the nature of the image accordingly when you create it. Or if the sector size of the reconstructed RAID or a disk in an enclosure does not match the sector size of the file system in a partition, you can adjust the sector size of the image accordingly. All of this will allow for smoother and more successful usage of the image later, in particular by users who do not pay much attention to details such as image type and sector size. With the additional metadata present for a raw image, X-Ways Forensics does not need to prompt users for the nature of the image and its sector size even if under normal circumstances it would (for example because the image does not start with an easily identifiable partitioning method or volume boot sector).

At the end of the imaging process, the computer can be optionally either shut down or (if supported by your system) hibernated, to save power. If you select hibernation and Windows signals that hibernation fails, X-Ways Forensics will instead try to shut down the system.

There is an option to add newly created images to the case and start refining their volume snapshot(s) automatically without further user interaction if the source disk had not been added to the case yet and if a case is open at that time when you start imaging.

Using this command is the recommended way to create a disk image. In order to image an arbitrary range of sectors, you could select a sector range as a block and copy it to a file via Edit | Copy Block | Into New File, or use Tools | Disk Tools | Clone Disk. The latter is particularly useful to partially image hard disks with severe physical defects (not just ordinary bad sectors) and can even copy sectors in reverse order.

It is also possible to image a physical device (e.g. local hard disk or remote hard disk or RAM opened through F-Response) automatically via the command line. The first parameter should start with a colon and then specify the number of the device in Windows (e.g. ":1" for hard disk No. 1, i.e. the second hard disk). This will cause that device to be opened automatically upon

start-up. The second parameter should start with a pipe, followed by either "e01" or "raw" to indicate the preferred image file format, followed by another pipe and the path and filename of the image (e.g. "|e01|G:\Output filename.e01"). The third parameter can be "auto" to automatically exit X-Ways Forensics after imaging.

The encryption algorithm optionally used in .e01 evidence files is either 128-bit or 256-bit AES/Rijndael, in counter (CTR) mode. This allows for random read access within evidence files. The 128-bit implementation is newer and faster and supported only by X-Ways Forensics v16.4 and later. This encryption algorithm uses a 256-bit key that is digested with SHA-256 from the 512-bit concatenation of the SHA-256 of the password you specify and 256 bits of cryptographically sound random input ("salt"), which is stored in the header of the evidence file. For 128-bit AES the 256-bit key is reduced to 128 bit by xor-ing the first and second half. The 128-bit counter is randomized and incremented per encryption block, as a little-endian integer in 256-bit AES, as a big-endian integer in 128-bit AES. The encryption block size of AES is 128 bits. An additional SHA-256 is stored in the header as well (optionally for 256-bit AES, see Security Options) and used later to determine whether a password, specified by the user for decryption, is correct or not. The SHA-256 algorithm is applied to a concatenation of the salt, hash *x*, and hash *y* to compute this password verification hash, where hash *x* is the SHA-256 of the user-supplied password and hash *y* is the SHA-256 of the concatenation of the user-supplied password and hash *x*. For 128-bit AES, *y* becomes *x* and is concatenated and hashed over and over again, 100,000 times, to practically render rainbow table attack computationally infeasible. Please note that when you use compression and encryption at the same time, each chunk in an .e01 evidence file is first compressed, then encrypted. So an educated guess about the *nature* of the data in a given chunk might be possible, merely judging from the compressed size of the chunk (i.e. its compression ratio), even if the compressed data is encrypted.

If you have WinHex assign a filename for a WinHex backup automatically, the file will be created in the folder for backups (cf. General Options), named with the next free "slot" according to the Backup Manager's naming conventions ("*xxx*.whx"), and will be available in the Backup Manager. If you explicitly specify a path and a filename, you can restore the backup or image later using the Restore Backup command, and in case of split backups WinHex will automatically append the segment number to the filenames.

10.8 Hints on Disk Cloning, Imaging, Image Restoration

Cloning or imaging with WinHex/X-Ways Forensics makes exact sector-wise, forensically sound copies, including all unused space and slack space. An image is usually preferable to a clone, as all data (and metadata such as timestamps) in an image file is protected from the operating system.

If you clone/image a disk for backup purposes, try to avoid that the disk is being written to by the operating system or other programs during the process, e.g. by unmounting partitions that are mounted as drive letters before starting. Such write operations are unavoidable, of course, if you clone/image the disk that contains the active Windows installation from where you execute WinHex/X-Ways Forensics. If the source disk is being written to during the process, the clone/image may have an inconsistent state from the point of view of the operating system (e.g. it

may not be able to boot a Windows installation any more). From a forensic standpoint, however, when cloning/imaging a live system, although it is highly desirable that no writing occurs any more, that should not be a major problem, as you still get an accurate snapshot of each and every sector.

If the destination of cloning or image restoration is a partition that is mounted as a drive letter, WinHex will try to clear all of Windows' internal buffers of that destination partition. If nonetheless you don't see the new contents in Windows Explorer on the destination after the operation has complete, you may simply need to reboot your system.

Note that WinHex does not dynamically change partition sizes and adapt partitions to destination disks larger or smaller than the source.

10.9 Skeleton Images

Forensic license only. A typical X-Ways feature that cements X-Ways Forensics' position as the tool that gives its users the greatest amount of control when selecting/targeting/filtering data at any conceivable level: The ability to create *forensic physical skeleton disk images* that contain only those sectors that are needed for certain purposes, while maintaining compatibility with other tools. These can be sectors with partition tables, file system data structures, their neighboring sectors as well as sectors with file contents or any sectors in unpartitioned no man's land. A skeleton image is typically sparsely populated with data, with vast areas in between remaining undefined, so that it makes sense to utilize NTFS sparse file technology for it. Unwritten areas in the skeleton image will act as if zeroed out when read later.

You start skeleton imaging by invoking the File | Create Skeleton Image menu command. Which sectors from then now will be copied into the image is defined indirectly, by making X-Ways Forensics *read* those sectors from the source disk that are needed for a certain purpose. When the target image is open in the background, next you typically open the disk or partition or open and interpret the image that you wish to acquire partially. That way it will be automatically defined as the source, and that way even read operations during the important opening or interpretation step are triggered already, when partition tables and boot sectors have to be parsed, so that these essential data structures that define partitions and identify file systems are included in the skeleton image.

So after opening a partitioned physical disk, you have a "basic skeleton" in your target image: Partition tables pointing to partition boot sectors or nested partition tables, whose function is to support all the other data in between (file system data and user data). If you also wish to ensure that from the skeleton image it is possible to take a volume snapshot of a certain partition, i.e. get a listing of all files and directories referenced by the file system in that partition, then you open that partition from the source hard disk so that a volume snapshot is actually taken. Again, all the sectors read from the source hard disk in the process are simultaneously copied to the image, and that is the file system data structures, e.g. $MFT in NTFS, all directory clusters in FAT, and the catalog file in HFS+. That adds considerably more administrative data and also metadata to your skeleton image, but still no or almost no user contents. Unrelated sectors that are not used by the file system are not read and therefore not copied. That also means that the ability to find

previously existing files in the skeleton image will be limited.

If you wish to include an arbitrary range of sectors in the image, you only need to find a way to make X-Ways Forensics read those sectors. For example, to include sectors from number 1,000,000 to 1,000,999, define those 1,000 sectors as a block and hash that block (in Disk mode) using the Tools | Compute Hash command, or run a physical search in that block only. Or, to acquire an unusually large partition gap between partition 1 and 2, you could hash the virtual file representing that gap. You can also manually navigate to any single sector of interest that you want to be included (e.g. Navigation | Go To Sector) or use any of the file system navigation menu commands. All of that works because reading sectors triggers their acquisition.

However, if you wish to specifically acquire selected *files*, that is easier, and it might be a good idea to turn off the indirect acquisition of any sectors that are read for whatever purpose along the way, so that for a example file that you preview and that turns out to be irrelevant is not acquired by the preview action already. For that, you can change the state of the skeleton image that is open in the background to "idle", using the State command in the File menu. In "idle" mode, only the "Add to [name of the skeleton image]" command in the directory browser context menu allows to acquire selected files (by temporarily activating the image and triggering read operations), .

If you wish to include some operating system files, for example, such as Windows registry hives, explore the partition recursively from the root directory, filter for those files and invoke the "Add to" command in the directory browser context menu. (Only available if no evidence file container is open in the background for filling at that time.) The examiner who only has the resulting skeleton image will consequently be able to view the hives and create a registry report about them, assuming you had already copied the file system data structures which are required to find out *which* sectors contain the data of the file.

The dialog window to change the state of the target image also allows you to close it, i.e. stop the acquisition for the moment or finalize the image. The same skeleton image can be further completed at any later time by selecting it again with the "Create Skeleton Image" command, but then you choose to not overwrite, but to update it.

As you see, you have full control over what data will make it into the image. The methology just assumes that you have some understanding of what data you want/need and, should that data not be stored in ordinary easy-to-select files, where to find it/how to get it physically. The sectors can be targeted in any order. Multiple reads of the same sectors don't change anything in the skeleton image and have no negative effect, except they may cause unnecessary duplicate lines in the optional log file that X-Ways Forensics can produce. Such a log file is created in the same directory as the skeleton image and will list all sector ranges that were copied, optionally along with the hash value of each sector range, which allows to manually verify the data in certain areas should there ever be doubt about it. If you use the "Add to" command to copy files to a skeleton image, the name of each such file will also be output in the log, followed by the sector ranges that correspond to to it (more than one if the file is fragmented or if X-Ways Forensics simply chooses to copy sectors in multiple chunks).

You may want to convert the resulting raw skeleton image into a compressed and/or encrypted .e01 evidence file and hash it or compress it with WinRAR or 7Zip etc. before passing it on to

other users. The compression rate will be unusually high if the skeleton image is only sparsely populated, and the speed of reading extremely high because undefined/unallocated areas do not have to be read from the disk. For your own use, you can just keep it as is since it does not use as much drive space as the nominal file size suggests thanks to NTFS sparse storage. If you wish to copy the raw skeleton image, be sure to copy it as a sparse file (can be done in X-Ways Forensics using the Tools | File Tools | Copy Sparse command) so that the copy will also be a sparse file and only takes as much drive space as the original file. A conventional copy command would copy even the vast unused and unallocated areas within the sparse file as binary zeroes.

To verify that the data transferred to a skeleton image has not changed, such an image can be hashed entirely, just like an ordinary image. Alternatively, and much quicker, you can use the command "Verify Skeleton Image" to hash only those sector ranges again that were actually transferred, according to the .log file (reading from the skeleton image), and compare the hash values to those in the .log file. Then, to verify that the .log file has not changed, it will be hashed itself, and the resulting highly valuable all encompassing master hash value is compared to the hash value stored in the optional .log.log file, if that file was created. It might be desirable to additionally verify that all unused areas in a skeleton image are still unallocated or at least filled with binary zeroes. This is not done by this function.

Options:

- A skeleton image should be created as an NTFS sparse file unless you intend to copy more than half of the sectors perhaps (just a very rough rule of thumb).
- If you don't have X-Ways Forensics set the nominal (logical) image file size to the full size of the source disk, then when interpreting the skeleton image and reading from it, a smaller "capacity" will be reported and you may get sector read errors. Still worth thinking about it for example if you wish to capture merely the first 1 MB of a 1 TB hard disk. Saves a lot of time if you wish to convert the skeleton image to an .e01 evidence file or want to hash it in its entirety.
- Skipping already zeroed out source sectors (sectors of the source disk that only contain binary zeroes) will treat such sectors exactly like sectors that were not acquired. This makes the resulting skeleton image smaller ("more sparse"), but it prevent you from showing with just the skeleton image that these sectors only contained zeroes on the source disk. They are indistinguishable from sectors that were not acquired.
- "Include directory data structures of the file system" has an effect when you apply the "Add to" command of the directory browser context menu to selected directories. If this option is selected, you will also copy the data structures of the file system for these directories, if there are any, e.g. INDX buffers in NTFS, subdirectory clusters in FAT, etc. (nothing in HFS+), otherwise only the contents of the files *in* these directories.
- "Report table associations" will create a report table association for every file that you specifically add to the skeleton image in the source volume snapshot, so that it is easy to see which files were copied already in case of any doubt.
- If "Create log file" is at least half checked, a .log file will be created that references all copied sector ranges. X-Ways Forensics makes an effort to prevent acquiring duplicate sectors, e.g. when copying the exact same sector range a second time or when copying overlapping sector ranges, so that can explain why you may not get more lines in the .log file when copying the same sectors again. If the checkbox is fully checked, a .log.log file about the .log file will be created with a hash of the .log file.

- All copied sector ranges can be optionally hashed, and the hash values can be written to the .log file and can be verified after closing the skeleton image.

Benefits of skeleton images:

- Partial image, saves drive space.
- Quick to create, especially when acquiring remote hard disks through a slow network connection using F-Response.
- Transports/reveals only specifically targeted data, excludes unrelated data, as may be required by law, common sense, time pressure or the customer.
- Ideally suitable for technical data structures (partition tables, file systems) and files in a file system as well.
- Ability to acquire all essential file system data without knowing anything about the file system and in which sectors its data structures are stored.
- Result works exactly like a conventional raw image of the disk for all the intended purposes if adequately prepared, with original offsets and relative distances between data structures preserved (unlike in an evidence file container).
- The file format is universal, and all forensic tools that support raw images have a chance to understand the data, unless they need more data than was included or already don't understand the partitioning method or file system etc. of the original complete disk/image.

Caveats:

- Note that a search hit list on the screen with context previews around the search hits for example will cause a lot of read activity, so you may want to change the state of the skeleton image to idle mode when it is open in the background in certain situations.
- To avoid that the start sectors of files or directories that you merely click in the directory browser in Partition/Volume mode are copied to the skeleton image (because such a click automatically jumps to the respective 1st sector), you can navigate the directory browser in Legend mode instead, or have to change the status of the image to "idle".
- Reading data from most *extracted* files such as e-mail messages, attachments, video stills, pictures embedded in MS Excel spreadsheets etc. do not trigger corresponding read operations at the disk level, so they cannot be copied. Skeleton images are suitable only for files at the file system level, not at any other level seen in volume snapshots. Use evidence file containers instead for such purposes.
- Note that to an unsuspecting examiner a skeleton image may look very much like an ordinary complete image. Such an examiner must be made aware of the incomplete, sparsely populated nature of the image. Unlike in a logical evidence file container, files whose contents are not contained in the image are not specially marked as such in a volume snapshot taken of an incomplete physical image. X-Ways Forensics v17.1 and later informs the examiner of the nature of an image when it's added to a case, if it detects a skeleton image.

A comparison of evidence file containers and skeleton images can be found on the web site.

Snippet imaging

A variant of skeleton imaging is called "snippet imaging". Click the button labelled "Snippet

imaging" in the file selection dialog of the File | Create Skeleton Image menu command to start snippet imaging. Any sectors that are being read by X-Ways Forensics from any disk or image while snippet imaging is active are written into separate files named after the sector number, with a .sector extension, in a subdirectory of the default directory for images named after the disk or volume. Contiguous sector reads are copied to a single file.

Snippet imaging mode can be deactivated by invoking the File | Snippet Imaging menu command. Snippet imaging is helpful in specific situations only, for example for debugging purposes, when in need for very specific sectors only that are best located by the software automatically (e.g. data structures needed when opening a particular file). Compared to skeleton imaging, snippet imaging can be beneficial because no image file of the same size as the source disk is created. (Even if it's a nominal size only and the image is sparse, sparse does not help if the file needs to be sent via Internet or copied to a file system that does not preserve the sparse nature of the file.)

Because of their compatible names, snippet image files can be directly used for sector superimposition. They can also conveniently and because of their typically small size very, very quickly be restored to a other disks, all such files in the same directory at the same time, of course taking the sector numbers in the filenames into account, by clicking the button "Snippet imaging" in the File | Restore Image dialog window.

10.10 Backup Manager

Displays a list of previously created WinHex backups. The items can be listed in a chronological or alphabetical order. Choose the backup you would like to restore. When that function completes, the original file or sector contents is shown.

You can restore the backup

- into a temporary file first such that you will still need to save it,
- directly and immediately to the disk, or
- to a new file.

In the case of disk sectors you may also wish to specify a different destination disk or a different destination sector number. It is also possible to only extract a subset of the sectors from the backup. (However, sectors at the beginning of a *compressed* backup cannot be left out during restoration.) If the backup was saved with a checksum and/or a digest, data authenticity is verified before the sectors will be directly written to the disk.

The backup manager also allows to delete backups which you do not need any longer. Backups that were created for internal use by the Undo command can be deleted by WinHex automatically (cf. Undo Options).

Backup files that are maintained by the backup manager are located in the folder specified in the General Options dialog. Their filenames are “*xxx*.whx” where *xxx* is a unique three-digit identification number. This number is displayed in the last column of the backup manager list.

10.11 Reconstructing RAID Systems

WinHex and X-Ways Forensics can internally destripe RAID level 0, 5, 5EE and 6 systems as well as JBOD consisting of up to 16 components. The components may be physical hard disks or images of physical disks for hardware RAIDs, or partitions for Linux software RAIDs. Components that are available as images need to be opened and interpreted before you use this function. You need to select the components in the correct order. WinHex lets you specify the stripe size in sectors (often 128 or at least a power of 2 like 32, 64, 256) and different RAID header sizes per component (often simply 0).

The header is a reserved area at the start of a component disk that some RAID controllers set aside for their private data and thus must be excluded from the reconstruction. If there are a few reserved sectors at the end of a component disk, as is not uncommon for JBOD, prior to the reconstruction you would specify the number of actually used sectors plus header size for each component via Tools | Disk Tools | Set Disk Parameters as the "Sector count".

You can usually tell that either the component order, the stripe size, the stripe pattern, or the RAID header size was selected incorrectly when no partitions are detected or partitions with unknown file systems or with file systems that cannot be interpreted properly.

When you add a reconstructed RAID system to a case (and optionally partitions opened from such a RAID system), the selected RAID configuration parameters are saved with the evidence object, which allows to access the RAID system instantly in later sessions (forensic licenses only).

In RAID level 5 and 6, data is not only striped across all component disks in a rotating pattern, but also interspersed with parity blocks for redundancy. RAID level 5 and 6 are implemented in different ways by different RAID controller manufacturers in that they employ different stripe/parity patterns. The supported patterns are the following:

```
Level 5: Backward Parity aka Left Asynchronous (Adaptec)
Component 1:  1  3  P
Component 2:  2  P  5
Component 3:  P  4  6

Level 5: Backward Dynamic Parity aka Left Synchronous (AMI and
Linux standard)
Component 1:  1  5  9  P
Component 2:  2  6  P  10
Component 3:  3  P  7  11
Component 4:  P  4  8  12

Level 5: Backward Delayed Parity (HP/Compaq)
Component 1:  1  3  5  7   9   11  13  15
Component 2:  2  4  6  8   P   P   P   P
Component 3:  P  P  P  P   10  12  14  16

Level 5: Forward Parity (aka Right Asynchronous)
```

```
Component 1:  P  3  5
Component 2:  1  P  6
Component 3:  2  4  P

Level 5: Forward Dynamic Parity (aka Right Synchronous)
Component 1:  P  6  8  10
Component 2:  1  P  9  11
Component 3:  2  4  P  12
Component 4:  3  5  7  P

Level 5: Forward Delayed Parity
Level 5: Forward Dynamic Delayed Parity (CRU/Dataport)

Level 5EE: Backward Parity (Adaptec)
Component 1:  1  3  S  P
Component 2:  2  S  P  7
Component 3:  S  P  5  8
Component 4:  P  4  6  S  (S = spare)

Level 5EE: Forward Parity
Component 1:  1  P  S  7
Component 2:  2  3  P  S
Component 3:  S  4  5  P
Component 4:  P  S  6  8

Level 6: Backward Parity (Adaptec/JetStor)
Component 1:  1  3  P  Q
Component 2:  2  P  Q  7
Component 3:  P  Q  5  8
Component 4:  Q  4  6  P

Level 6: Backward Dynamic Parity
Component 1:  1  4  P  Q
Component 2:  2  P  Q  7
Component 3:  P  Q  5  8
Component 4:  Q  3  6  P

Level 6: Forward Delayed Parity
Level 6: Forward Parity
```

The parity start component can be defined differently if necessary, for many RAID variants. To stick with the select standard pattern, leave that value at 0. In order to define a non-standard parity start component, specify the number of the component where the parity is located first (1-based).

The delay with that the parity moves on HP/Compaq controllers is most often 4 or 16, but freely configurable.

If one of the RAID component disks is not available, you can reconstruct a RAID 5 system nonetheless because one component is redundant. Simply select a dummy substitute (one of the *other, available* components of the same RAID system) as the *missing* component and declare

that component “missing”! RAID 5EE and RAID 6 can also be internally reconstructed if one component is missing.

Appendix A: Template Definition

1 Header

The header of a template definition has the following format:

```
template "title"
[description "description"]
[applies_to (file/disk/RAM)]
[fixed_start offset]
[sector-aligned]
[requires offset "hex values"]
[big-endian]
[hexadecimal/octal]
[read-only]
[multiple [fixed overall size]]
// Put any general comments to the template here.
begin
      variable declarations
end
```

Tags in brackets are optional. The order of the tags is irrelevant. Expressions must only be enclosed in inverted commas if they contain space characters. Comments may appear anywhere in a template definition. Characters following a double slash are ignored by the parser.

The keyword `applies_to` must be followed by one and only one of the words `file`, `disk`, or `RAM`. WinHex issues a warning if you are going to use a template on data from a different source.

While by default templates start interpreting the data at the current cursor position when applied, an optional `fixed_start` statement ensures interpretation always starts at the specified absolute offset within the file or disk.

If the template applies to a disk, the keyword `sector-aligned` ensures the template interpretation starts at the beginning of the current sector, regardless of the exact cursor position.

Similar to the `applies_to` statement, the `requires` statement enables WinHex to prevent an erroneous application of a template definition to data that does not match. Specify an offset and a hex-value chain of an arbitrary length that identifies the data for which the template definition was intended. For example, a valid master boot record can be recognized by the hex values 55 AA at offset 0x1FE, an executable file by the hex values 4D 5A (“MZ”) at offset 0x0. There may be multiple `applies_to` statements in a template definition header, which are all considered.

The keyword `big-endian` causes all multi-byte integer and boolean variables in the template definition to be read and written in big-endian order (high-order byte first).

The keyword `hexadecimal` causes all integer variables in the template definition to be displayed in hexadecimal notation.

The keyword `read-only` ensures that the template can only be used to examine, but not to manipulate data structures. The edit controls within the template will be grayed out.

If the keyword `multiple` is specified in the header, WinHex allows browsing to neighboring data records while displaying the template. This requires that WinHex has knowledge of the record's size. If it is not specified as a parameter to the `multiple` statement, WinHex assumes the overall size of a template structure (=record) to be the current position at the end of the template interpretation less the base editing position. If this is a variable size, i.e. array sizes or `move` parameters are determined dynamically by the value of variables, WinHex cannot browse to precedent data records.

2 Body: Variable Declarations

The body of a template definition mainly consists of variable declarations, similar to those in programming languages. A declaration has the basic form

```
type "title"
```

where `type` can be one of the following:

- `int8, uint8 = byte, int16, uint16, int24, uint24, int32, uint32, uint48, int64,`
- `uint_flex,`
- `binary,`
- `float = single, real, double, longdouble = extended,`
- `char, char16, string, string16,`
- `zstring, zstring16,`
- `boole8 = boolean, boole16, boole32,`
- `hex,`
- `DOSDateTime, FileTime, OLEDateTime, SQLDateTime, UNIXDateTime = time_t, JavaDateTime,`
- `GUID`

`title` must only be enclosed in inverted commas if it contains space characters. `title` must not consist only of digits. WinHex does not distinguish between upper and lower case characters in titles. 41 characters are used to identify a variable at most.

`type` can be preceded by at most one member of each of the following modifier groups:

```
big-endian      little-endian
hexadecimal     decimal         octal
read-only       read-write
local
```

These modifiers only affect the immediately following variable. They are redundant if they appear in the header already. "local" translates timestamps except DOSDateTime from UTC to the timezone specified in the General Options.

The number at the end of a type name denotes the size of each variable (strings: of each character) in bits. With `char16` and `string16`, WinHex supports Unicode characters and strings. However, Unicode characters other than the first 256 ANSI-equivalent characters are not supported. The maximum string size that can be edited using a template is 8192 bytes.

The types `string`, `string16`, and `hex` require an additional parameter that specifies the number of elements. This parameter may be a constant or a previously declared variable. If it is a constant, it may be specified in hexadecimal format, which is recognized if the number is preceded by `0x`.

You may declare arrays of variables by placing the array size in square brackets next to the type or the title. Specify "unlimited" as the array size to make the template stop only when the end of file is encountered. The following two lines declare a dynamically sized ASCII string, whose length depends on the preceding variable:

```
uint8      "len"
char[len]  "A string"
```

The same could be achieved by the following two declarations:

```
byte       "len"
string len "A string"
```

The character "~" can be used as placeholder for later replacement with the actual array element number (see below). This does not apply to arrays of `char` variables, since they are automatically translated into a string.

Numerical parameters of `string`, `string16`, and hex variables as well as array size expressions may be specified in mathematical notation. They will be processed by the integrated formula parser. Such expressions need to be enclosed in brackets. They must not contain space characters. They may make use of previously declared integer variables whose names do not contain space characters either. Supported operations are addition (+), subtraction (-), multiplication (*), integer division (/), modular division (%), bitwise AND (&), bitwise OR (|), and bitwise XOR (^). Valid mathematical expressions are for example (5*2+1) or (len1/(len2+4)). The result is always an integer and must be a positive number.

`zstring` and `zstring16` are null-terminated strings whose size is determined dynamically at run-time.

3 Body: Advanced Commands

When enclosed in braces, several variable declarations comprise a block that can be used

repeatedly as a whole. Note, however, that blocks must not be *nested* in the current implementation. The character ~ can be used in a variable's name as a placeholder for later replacement with the actual repetition count. The optional `numbering` statement defines where to begin counting (0 by default).

```
numbering  1
{
byte       "len"
string len "String No. ~"
}[10]
```

In this example the actual variable names in the template will be "String No. 1", "String No. 2", ..., "String No. 10". Instead of a constant number of repetitions (10 in this example), you may also specify "unlimited". In that case WinHex will repeat the block until the end of file is encountered. "ExitLoop" can be used to break out of a loop at any time. "Exit" terminates execution of the template completely.

"IfEqual" is useful for the comparison of two expressions. Operands can be either both numerical values, be it constant values in decimal notation, integer variables or a formulas, or byte sequences given as text or hex values which are compared byte by byte. ASCII string expressions must be enclosed in quotation marks, hex sequences must be preceded by a "0x" identifier. Formulas need to be enclosed in brackets.

```
{
byte       Value
IfEqual    Value 1
     ExitLoop
EndIf
} [10]
```

An "IfEqual" command block is terminated with an "EndIf" statement. If the compared expressions are equal, template interpretation continues after "IfEqual". Optionally, "IfEqual" can be followed by an "Else" statement. The template processor branches into the "Else" block if the expressions are not equal. "IfEqual" commands must not be nested. "IfGreater" is similar to "IfEqual". The condition is true if the first expression is greater than the second. Strings and hex values are compared lexicographically.

In order to facilitate reading and navigating the template, you may define groups of variables that are separated by empty space in the dialog box:

```
section    "...Section Title..."
...
endsection
```

The `section`, `endsection`, and `numbering` statements do not advance the current position in the data to be interpreted.

There are two commands that do not declare variables either, but are explicitly used to change the current position. This can be done to skip irrelevant data (forward movement) or to be able access certain variables more than once as different types (backward movement). Use the `move n`

statement to skip n bytes from the current position, where n may be negative. `goto n` navigates to the specified absolute position from the beginning of the template interpretation (must be positive). `gotoex n` jumps to the specified absolute position based on the start of the data window (e.g. file or disk).

The following example demonstrates how to access a variable both as a 32-bit integer and as a four-part chain of hex values:

```
int32      "Disk serial number (decimal)"
move -4
hex 4      "Disk serial number (hex)"
```

4 Body: Flexible Integer Variables

A special variable type supported by templates is `uint_flex`. This type allows to compose an unsigned integer value from various individual bits within a 32-bit (4-byte) range in an arbitrary order and is even more flexible than a so-called bit field in the C programming language.

uint_flex requires an additional parameter string in inverted commas that specifies exactly which bits are used in which order, separated by commas. The bit listed first becomes the most significant bit (high value bit) in the resulting integer, and it is not interpreted as a + or - indicator. The bit listed last becomes the least significant bit in the resulting integer.

The bits are counted starting with 0. Bit 0 is the bit that is the least significant bit of the 1st byte. Bit 31 is the most significant bit of the fourth byte. Thus, the definition is based on little-endian philosophy.

For example,
`uint_flex "15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0" "Standard 16-bit integer"`
is exactly the same as `uint16`, the common unsigned 16-bit integer variable.

`uint_flex "31,30,29,28,27,26,25,24,23,22,21,20,19,18,17,16,15,14,13,12, 11,10,9,8,7,6,5,4,3,2,1,0" "Standard 32-bit integer"`
is exactly the same as `uint32`, the common unsigned 32-bit integer variable.

The benefit of `uint_flex`, though, is that the number, the position, and the usage order of all bits can be chosen arbitrarily. For example,
`uint_flex "7,15,23,31" "An unusual 4-bit integer"`
composes a 4-bit integer out of the respective most significant bits of each of the four bytes involved. If these four bytes happen to be
F0 A0 0F 0A =
11110000 **1**0100000 **0**0001111 **0**0001010,
bit 7 is **1**, bit 15 is **1**, bit 23 is **0**, and bit 31 is **0**.
So the resulting uint_flex is **110**0 = **1***8 + **1***4 + **0***2 + **0***1 = 12.

Appendix B: Script Commands

Script commands are case-*in*sensitive. Comments may occur anywhere in a script file and must be preceded by two slashes. Parameters may be 255 characters long at most. Where in doubt because hex values, text strings (or even integer numbers) are accepted as parameters, you may use inverted commas (quotation marks) to enforce the interpretation of a parameter as text. Inverted commas are *required* if a text string or variable name contains one or more space characters, so that all characters between the inverted commas are recognized as constituting *one* parameter.

Wherever numerical parameters are expected (integer numbers), the integrated formula parser allows you to use mathematical expressions. Such expressions need to be enclosed in brackets. They must not contain space characters. They may make use of variables that can be interpreted as integer numbers. Supported operations are addition (+), subtraction (-), multiplication (*), integer division (/), modular division (%), bitwise AND (&), bitwise OR (|), and bitwise XOR (^). Valid mathematical expressions are for example (5*2+1), (MyVar1/(MyVar2+4)), or (-MyVar).

The following is a description of currently supported script commands, including example parameters.

Create "D:\My File.txt" 1000
Creates the specified file with an initial file size of 1000 bytes. If the file already exists, it is overwritten.

Open "D:\My File.txt"
Open "D:*.txt"
Opens the specified file(s). Specify "?" as the parameter to let the user select the file to open.

Open C:
Open D:
Opens the specified logical drive. Specify ":?" as the parameter to let the user select a logical drive or physical disk to open.

Open 80h
Open 81h
Open 9Eh
Opens the specified physical media. Floppy disk numbering starts with 00h, fixed and removable drive numbering with 80h, optical media numbering with 9Eh.

Optionally, you may pass a second parameter with the Open command that defines the edit mode in which to open the file or media ("in-place" or "read-only").

CreateBackup
Creates a WHX backup of the active file in its current state.

CreateBackupEx 0 100000 650 true "F:\My backup.whx"
Creates a WHX backup of the active disk, from sector 0 through sector 1,000,000. The backup file will be split automatically at a size of 650 MB. Compression is enabled (“true”). The output file is specified as the last parameter.
If the backup file should not be split, specify 0 as the third parameter. To disable compression, specify “false”. To have the Backup Manager automatically assign a filename and place the file in the folder for backup files, specify "" as the last parameter.

Goto 0x128
Goto MyVariable
Moves the current cursor position to the hexadecimal offset 0x128. Alternatively, an existing variable (up to 8 bytes large) can be interpreted as a numeric value, too.

Move -100
Moves the current cursor position 100 bytes back (decimal).

Write "Test"
Write 0x0D0A
Write MyVariable
Writes the four ASCII characters "Test" or the two hexadecimal values "0D0A" at the current position (in overwrite mode). Can also write the contents of a variable specified as the parameter. Moves the current position forward by the number of bytes written. When the end of the file is reached, to accomplish that, a null byte is appended. Useful so that further Write commands don't overwrite the last byte written by the previous Write command.

Write2
Identical to Write, but does not append a null byte if the end of the file has been reached. So it is not safe to assume that Write2 always moves the current position forward by the number of bytes written.

Insert "Test"
Functions just as the "Write" command, but in *insert* mode. Must only be used with *files*.

Read MyVariable 10
Reads the 10 bytes from the current position into a variable named "MyVariable". If this variable does not yet exist, it will be created. Up to 48 different variables allowed. Another way to create a variable is the Assign command.

ReadLn MyVariable
Reads from the current position into a variable named "MyVariable" until the next line break is encountered. If the variable already exists, its size will be adjusted accordingly.

Close
Closes the active window without saving.

CloseAll
Closes all windows without saving.

Save
Saves changes to the file or disk in the active window.

SaveAs "C:\New Name.txt"
Saves the file in the active window under the specified path. Specify "?" as the parameter to let the user select the destination.

SaveAll
Saves changes in all windows.

Terminate
Aborts script execution.

Exit
Terminates script execution and ends WinHex.

ExitIfNoFilesOpen
Aborts script execution if no files are already opened in WinHex.

Block 100 200
Block "My Variable 1" "My Variable 2"
Defines the block in the active window to run from offset 100 to offset 200 (decimal). Alternatively, existing variables (each up to 8 bytes large) can be interpreted as numeric values.

Block1 0x100
Defines the block beginning to be at the hexadecimal offset 0x100. A variable is allowed as the parameter as well.

Block2 0x200
Defines the block end to be at the hexadecimal offset 0x200. A variable is allowed as the parameter as well.

Copy
Copies the currently defined block into the clipboard. If no block is defined, it works as known from the Copy command in the Edit menu.

Cut
Cuts the currently defined block from the file and puts it into the clipboard.

Remove
Removes the currently defined block from the file.

CopyIntoNewFile "D:\New File.dat"
CopyIntoNewFile "D:\File +MyVariable+.dat"
Copies the currently defined block into the specified new file, without using the clipboard. If no block is defined, it works as known from the Copy command in the Edit menu. Can copy disk

sectors as well as files. The new file will not be automatically opened in another edit window. Allows an unlimited number of "+" concatenations in the parameter. A variable name will be interpreted as an integer if not be larger than 2^24 (~16 Mio.). Useful for loops and file recovery.

Paste
Pastes the current clipboard contents at the current position in a file, without changing the current position.

WriteClipboard
Writes the current clipboard contents at the current position in a file or within disk sectors, without changing the current position, by overwriting the data at the current position.

Convert *Param1 Param2*
Converts the data in the active file from one format into another one. Valid parameters are ANSI, IBM, Binary, HexASCII, IntelHex, MotorolaS, Base64, UUCode, LowerCase, UpperCase, and hiberfil,, in combinations as known from the Convert menu command.

AESEncrypt "My Password"
Encrypts the active file or disk, or selected block thereof, with the specified key (up to 32 characters long) with AES.

AESDecrypt "My Password"
Decrypts the active file or disk.

Find "John" [*MatchCase MatchWord Down Up BlockOnly SaveAllPos Unicode Wildcards*]
Find 0x1234 [*Down Up BlockOnly SaveAllPos Wildcards*]
Searches in the active window for the name John or the hexadecimal values 0x1234, respectively, and stops at the first occurrence. Other parameters are optional. By default, WinHex searches the entire file/disk. The optional parameters work as known from usual WinHex search options.

ReplaceAll "Jon" "Don" [*MatchCase MatchWord Down Up BlockOnly Unicode Wildcards*]
ReplaceAll 0x0A 0x0D0A [*Down Up BlockOnly Wildcards*]
Replaces all occurrences of either a string or hexadecimal values in the active file with something else. Can only be applied to a disk if in in-place mode.

IfFound
A boolean value that depends on whether or not the last Find or ReplaceAll command was successful. Place commands that shall be executed if something was found after the IfFound command.

IfEqual MyVariable "Hello World"
IfEqual 0x12345678 MyVariable
IfEqual MyVariable 1000
IfEqual MyVariable MyOtherVariable
IfEqual MyVariable (10*MyOtherVariable)
Compares either two numerical integer values (each of them being a constant value, an integer variable or a mathematical expression) or two variables, ASCII strings, or hexadecimal values at

the binary level. Comparing two objects at the binary with a different length always returns False as the result. If equal, the following commands will be executed. If conditions must not be nested.

IfGreater MyVariable "Hello World"
IfGreater 0x12345678 MyVariable
IfGreater MyVariable 1000
IfGreater MyVariable MyOtherVariable
IfGreater MyVariable (10*MyOtherVariable)
Accepts the same parameters as IfEqual. If the first one is greater than the second one, the following commands will be executed. If conditions must not be nested.

Else
May occur after IfFound or IfEqual. Place commands that shall be executed if nothing was found or if the compared objects are not equal after the Else command.

EndIf
Ends conditional command execution (after IfFound, IfEqual, IfGreater).

{...
ExitLoop
...}
Exits a loop. A loop is defined by braces. Closing braces may be followed by an integer number in square brackets, which determines the number of loops to execute. This is may also be a variable or the keyword "unlimited" (so the loop can only be terminated with an ExitLoop command). Loops must not be nested.

Example of a loop:
{ Write "Loop" }[10] will write the word "Loop" ten times.

Label ContinueHere
Creates a label named "ContinueHere"

JumpTo ContinueHere
Continues script execution with the command following that label.

NextObj
Switches cyclically to the next open window and makes it the "active" window. E.g. if 3 windows are open, and window #3 is active, NextObj will make #1 the active window.

ForAllObjDo
The following block of script commands (until **EndDo** occurs) will be applied to all open files and disks.

CopyFile C:\A.dat D:\B.dat
Copies the contents of C:\A.dat into the file D:\B.dat.

MoveFile C:\A.dat D:\B.dat

Moves the file C:\A.dat to D:\B.dat.

DeleteFile C:\A.dat
Surprisingly, deletes C:\A.dat.

InitFreeSpace
InitSlackSpace
Clears free space or slack on the current logical drive, respectively, using the currently set initialization settings. InitSlackSpace switches the drive temporarily to in-place mode, thus saving all pending changes.

InitMFTRecords
Clears unused MFT FILE records on the current logical drive if it is formatted with NTFS, using the currently set initialization settings. Simply does nothing on other file systems. The changes are written immediately to the disk.

Assign MyVariable 12345
Assign MyVariable 0x0D0A
Assign MyVariable "I like WinHex"
Assign MyVariable MyOtherVariable
Stores the specified integer number, binary data, ASCII text, or other variable's contents in a variable named "MyVariable". If this variable does not yet exist, it will be created. Other ways to create variables: e.g. Read, GetUserInput, InttoStr. Up to 48 different variables allowed to exist simultaneously.

Release MyVariable
Specifically disposes an existing variable. Mandatory to invoke only when more than 48 variables with different names are to be used during the execution of a script, so that earlier variables that are not needed any more can be destroyed.

SetVarSize MyVariable 1
SetVarSize MyVariable 4
Explicitly sets the allocated memory size of a variable at a given time, in bytes. This can be useful e.g. for variables that hold integer values and that are the result of a calculation, if this value is to be written to a binary file with a fixed-length structure. Without SetVarSize, no assumption must be made about the size of the variable. For instance, the number 300 could be stored in any number of bytes larger than 1. If the new size set by SetVarSize is smaller than the old size, the allocated memory is truncated. If the new size is larger, the allocated memory is expanded. At any rate, the value of the persisting bytes is retained.

GetUserInput MyVariable "Please enter your name:"
Stores the ASCII text or binary data (0x...) specified by the user at script execution time (128 bytes at max.) in a variable named "MyVariable". The user is prompted by the message you provide as the second parameter. If the variable does not yet exist, it will be created. Other ways to create variables: Assign, Read.

GetUserInputI MyIntegerVariable "Please enter your age:"

Works like GetUserInput, but accepts and stores only integer numbers.

Inc MyVariable
Interprets the variable as an integer (if not larger than 8 bytes) and increments it by one. Useful for loops.

Dec MyVariable
Interprets the variable as an integer (if not larger than 8 bytes) and decrements it by one.

IntToStr MyStr MyInt
IntToStr MyStr 12345
Stores the decimal ASCII text representation of the integer number specified as the second parameter in a variable specified as the first parameter.

StrToInt MyInt MyStr
Stores the binary representation of the integer number specified as a decimal ASCII string in the second parameter in a variable specified as the first parameter.

StrCat MyString MyString2
StrCat MyString ".txt"
Appends one string to another. The second parameter may be a variable or a constant string. The first parameter must be a variable. The result will be saved in the variable specified by the first parameter and must not be longer than 255 characters.

GetClusterAlloc MyStr
May be applied to a logical volume. Retrieves a textual description of the current position's allocation, e.g. which file is stored in the current cluster, and saves that description in the specified variable.

GetClusterAllocEx IntVar
May be applied to a logical volume. Retrieves an integer value that indicated whether the cluster at the current position is allocated (1) or not (0), and saves that description in the specified variable.

GetClusterSize IntVar
May be applied to a logical volume. Retrieves the cluster size and saves that value in the specified integer variable.

InterpretImageAsDisk
Treats a raw image or evidence file like the original physical disk or partition. Requires a specialist or forensic license.

CalcHash HashType MyVariable
CalcHashEx HashType MyVariable
Calculates a hash as known from the command in the Tools menu and stores it in the specified variable (which will be created if it does not yet exist). The HashType parameter must be one of the following: CS8, CS16, CS32, CS64, CRC16, CRC32, MD5, SHA-1, SHA-256, PSCHF.

CalcHashEx in addition displays the hash in a dialog window.

MessageBox "Caution"
Displays a message box with the text "Caution" and offers the user an OK and a Cancel button. Pressing the Cancel button will abort script execution.

ExecuteScript "ScriptName"
Executes another script from within a running script, at the current execution point, e.g. depending on a conditional statement. Calls to other scripts may be nested. When the called script is finished, execution of the original script will be resumed with the next command. This feature can help you structure your scripts more clearly.

Turbo On
Turbo Off
In turbo mode, most screen elements are not updated during script execution and you are not able to abort (e.g. by pressing Esc) or pause. This may accelerate script execution if a lot of simple commands such as Move and NextObj are executed in a loop.

Debug
All the following commands must be confirmed individually by the user.

UseLogFile
Error messages are written into the log file "Scripting.log" in the folder for temporary files. These messages are not shown in a message box that requires user interaction. Useful especially when running scripts on unattended remote computers.

CurrentPos
GetSize
unlimited
are keywords that act as a placeholders and may be used where numeric parameters are required. On script execution, CurrentPos stands for the current offset in the active file or disk window and GetSize for its size in bytes. unlimited actually stands for the number 2,147,483,647.

Appendix C: Master Boot Record

The Master Boot Record is located at the physical beginning of a hard disk, editable using the disk editor. It consists of a master bootstrap loader code (446 bytes) and four subsequent, identically structured partition records. Finally, the hexadecimal signature 55AA completes a valid Master Boot Record.

The format of a partition record is as follows:

Offset	Size	Description
0	8 bit	A value of 80 designates an active partition.
1	8 bit	Partition start head

2	8 bit	Partition start sector (bits 0-5)
3	8 bit	Partition start track (bits 8,9 in “start sector” as bits 6,7)
4	8 bit	Operating system indicator, see below
5	8 bit	Partition end head
6	8 bit	Partition end sector (bits 0-5)
7	8 bit	Partition end track (bits 8,9 in “end sector” as bits 6,7)
8	32 bit	Sectors preceding partition
C	32 bit	Length of partition in sectors

Operating system indicators:
(hexadecimal, incomplete list)

00	Empty partition-table entry
01	DOS 12-bit FAT
04	DOS 16-bit FAT (up to 32M)
05	DOS 3.3+ extended partition
06	DOS 3.31+ Large File System (16-bit FAT, over 32M)
07	Windows NT NTFS, OS/2 HPFS, Advanced Unix
08	OS/2 v1.0-1.3, AIX bootable partition, SplitDrive
09	AIX data partition
0A	OS/2 Boot Manager
0B	Windows 95 with 32-bit FAT
0C	Windows 95 with 32-bit FAT (using LBA-mode INT 13 extensions)
0E	Logical-block-addressable VFAT (same as 06, but using LBA-mode INT 13)
0F	Logical-block-addressable VFAT (same as 05, but using LBA-mode INT 13)
17	Hidden NTFS partition
1B	Hidden Windows 95 FAT32 partition
1C	Hidden Windows 95 FAT32 partition (using LBA-mode INT 13 extensions)
1E	Hidden LBA VFAT partition
42	Dynamic disk volume
50	OnTrack Disk Manager, read-only partition
51	OnTrack Disk Manager, read/write partition
81	Linux
82	Linux Swap partition, Solaris (Unix)
83	Linux native file system (ext2fs/xiafs)
84	Hibernation partition
85	Linux EXT
86	FAT 16 volume/stripe set (Windows NT)
87	HPFS fault-tolerant mirrored partition, NTFS volume/stripe set
A0	Laptop hibernation partition
BE	Solaris boot partition
C0	DR-DOS/Novell DOS secured partition
C6	Corrupted FAT 16 volume/stripe set (Windows NT)
C7	Corrupted NTFS volume/stripe set
DE	DELL OEM partition

F2	DOS 3.3+ secondary partition
FE	IBM OEM partition