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

Copyright © 1995-2011 Stefan Fleischmann, X-Ways Software Technology AG. All rights reserved.

Contents

1 Preface

1.1 About WinHex and X-Ways Forensics

Copyright © 1995-2011 Stefan Fleischmann, X-Ways Software Technology AG. All rights reserved.

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 v16.0, released April 2011. It is available in English and German.

Supported operating systems: Windows 2000, Windows XP, Windows 2003 Server, Windows Vista/2008 Server, Windows 7. 32-bit and 64-bit.

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.

Professional users around the world include...
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. Please visit the web site to find out how to order the full version!

User interface translation: Chinese by Sprite Guo. Japanese by 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).

1.2 Legalities

Copyright © 1995-2011 Stefan Fleischmann, X-Ways Software Technology AG. No part of this publication may be reproduced, or stored in a database or retrieval system without the prior permission of the author. Any brand names and trademarks mentioned in the program or in this manual are properties of their respective holders and are generally protected by laws.

This publication is designed to provide accurate and authoritative information in regard to the subject matter covered. However, the author neither offers any warranties or representations nor does he accept any liability with respect to the program or the manual.

License Agreement

Your use, distribution, or installation of a software product published by X-Ways Software Technology AG indicates your acceptance of this license agreement. If you do not agree to any of the terms, then do not install, distribute or use the product.

A trial version may be only used for evaluation purposes. Purchasing one license authorizes you to install one copy of the full version of the software on a single machine at a time, usage by one person at a time. Additional licenses authorize you to install and use the full version on additional machines at the same time or to have it used by multiple persons at the same time. Exception: For computers in the same location, *forensic* licenses for WinHex/X-Ways Forensics do not impose an upper limit on the number of computers with *installations* of the software, only on the number of concurrent uses on different computers.

Applies to products for that an evaluation version exists: The software, and all accompanying files, data, and materials, are distributed “as is” and with no warranties of any kind, whether express or implied, to the maximum extent permitted by applicable law.

The user must assume the entire risk of using the program, knowing in particular that this software is not designed or intended for use in hazardous environments requiring fail-safe performance, where its failure to perform, misuse or inability to use adequately can reasonably be expected to lead to death, personal injury, or severe physical or environmental damage. In no event shall X-Ways Software Technology AG, or its officers, directors, employees, affiliates, contractors, or subsidiaries be liable for any direct, indirect, incidental, consequential, or punitive damages whatsoever arising out of the use or inability to use the software, to the maximum extent permitted by applicable law. Any liability will be limited exclusively to refund of purchase price by X-Ways Software Technology AG. It's the responsibility of the user to back up all data at reasonable intervals to minimize the damage caused by data losses of any kind.

You may not rent, lease, modify, translate, reverse-engineer, decompile or disassemble the software or create derivative works based on it without prior explicit permission. All rights of any kind in the software product which are not expressly granted in this license agreement are entirely

and exclusively reserved to and by X-Ways Software Technology AG.

No component of the software (except the WinHex API) must be accessed by other applications or processes.

Should any part of this agreement be or become invalid, such invalidity shall not affect the validity of the remaining provisions of the agreement.

Acknowledgements

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

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.

Outside In® Viewer Technology © 1991, 2007, 2008 Oracle.

NEXT3® is a registered trademark of CTERA Networks.

X-Ways Forensics contains an unofficial build of DevIL. DevIL is governed by the LGPL (http://www.gnu.org/copyleft/lesser.html), version 2.1. The original souce code can be downloaded from http://openil.sourceforge.net/.

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 and reverse disk cloning/imaging. Particularly useful for IT security specialists. Plus X-Ways Replica 1.3, a DOS-based forensically sound disk cloning and imaging software is included.

- Forensic licenses 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+, and UFS 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. When purchasing a forensic license, you will receive a dongle that allows to run the software. Also includes X-Ways Replica 2.36, with advanced disk cloning and imaging capabilities under DOS.

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 and X-Ways Forensics share the same code base. X-Ways Forensics offers numerous additional features over WinHex with a license. With a license for X-Ways Forensics, you can alternatively also use WinHex with the same license (and the same dongle). Both programs then offer the same full forensic feature set and are identical except for the following:

- WinHex (winhex.exe) always identifies itself as WinHex in the user interface, X-Ways Forensics (xwforensics.exe) as X-Ways Forensics. The program help and the manual, however, statically refer to “WinHex” in most cases.

- winhex.exe is available as a separate download for users of X-Ways Forensics as an add-on. When adding winhex.exe to an X-Ways Forensics installation, do not mix different versions of winhex.exe and X-Ways Forensics.

- 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) then you could run WinHex instead. With WinHex you can edit disk sectors and wipe entire hard disks, free space, or slack space.

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

1.5 Getting Started with X-Ways Forensics

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. The indexing feature can be found in the Search menu, too. 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. 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.

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.

- **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 tored in big endian, which is the typical byte order in Java, but in little endian in BlackBerry memory.

2.6 ANSI ASCII/IBM ASCII

ANSI ASCII is the character set used in non-Unicode Windows applications. It is standardized by the American National Standards Institute. MS-DOS uses the IBM ASCII character set (also called OEM character set). These character sets differ in the second half, containing characters with a ASCII values greater than 127. It is reasonable to select “IBM ASCII” in the View menu only when viewing or editing files originating from a DOS program.

Use the “Convert” command of the Edit menu to convert text files from one character set into 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

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.

2.8 Digests

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, and ed2k (specialist and forensic licenses only).

2.9 Technical Hints

- Technical specifications

Supported disk and file size:	at least 16 TB
Maximum sector number:	2^{40}-1
Maximum cluster number:	2^{32}-1, stellenweise 2^{31}-1
File systems support for volumes with more than 2^{32} sectors:	NTFS only
Maximum number of windows:	1000 (Win2K/XP/Va), 500 (Win9x/Me)
Maximum number of parallel program instances:	99
Maximum number of reversible keyboard inputs:	65535
Encryption depth:	128-256 bit
Digest length in backups:	128/256 bit
Character sets supported:	ANSI ASCII, IBM ASCII, EBCDIC, Unicode (UCS-2LE)
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.

- The user interface looks best if *no* extra large font is used in your Windows system.

- WinHex expects your computer to be running in little-endian mode.

- 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.

3 Forensic Features

3.1 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).

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 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.

Multi-examiner support for large cases

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.

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.

3.2 Evidence Objects

You may add any currently attached computer medium (such as hard disk, memory card, USB stick, CD-ROM, DVD, ...), any image file, or ordinary 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.

In the evidence object properties window, you may enter a title or number for that evidence object according to your own conventions. 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 WinHex automatically (as known from the Medial Details Report command in the Specialist menu). You may have WinHex calculate a hash (checksum or digest) on the evidence object and verify it 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.

Ways how to add files or media to a case: The "Add" commands in the case data window's File menu. The "Add" command in the edit window's tab's context menu. The "Add" command of a directory browser's item's context menu.

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.

3.3 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.

3.4 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 headere line, an optional logo, an optional preface, 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.
- Report tables: All files in selected report tables will 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.
- Case log

By default, the report is created for the entire case. Optionally it is created for selected evidence objects only.

3.5 Volume Snapshots

A volume snapshot is a database of the contents of a volume (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, ...), just 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 as a set of files named Volume*.dir, either in the folder for temporary files or (if associated with a case) in the evidence object's metadata directory.

The Specialist menu allows to *expand/refine* the standard volume snapshot in various ways. Requires a specialist or forensic license.

Particularly thorough file system data structure search

- FAT12/FAT16/FAT32: Searches for orphaned subdirectories (subdirectories that are no longer referenced by any other directory).
- NTFS: Searches for FILE records in sectors that do not belong to the current MFT. Such FILE records can be found e.g. in free space after a partition has been recreated, reformatted, moved, resized, or defragmented. They can also be found in volume shadow copies, and if so, will be interpreted if they refer to an older version of already known files that have changed, or to files that are totally unknown in the current volume snapshot. Old files found in shadow copies are marked with (SC) in the Attr. column, which makes them filterable.
- NTFS: With a forensic license, in a second and third step, this function also searches INDX buffers and $LogFile for noteworthy index record remnants, which 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).
- 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.
- Other: no difference

Taking a *thorough* volume snapshot is possibly a lengthy operation, depending on the size of the volume, and for that reason this is not the standard procedure when opening volumes.

The "**File header signature search**" option 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 signatures, and always at sector boundaries, because the partitions are treated as separate, additional evidence objects.

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 the 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.

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.

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 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.

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. Office documents can be Zip archives, too, and if so will be processed in the same way.
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.

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), Outlook Message (.msg), Outlook Template (.oft), Outlook Express (versions 4, 5, and 6, .dbx), Kerio Connect (store.fdb files that can be processed like PST/OST files), AOL PFC files, Mozilla mailbox (including Netscape and Thunderbird), generic mailbox (mbox, Berkeley mail format, BSD mail format, Unix mail format), Eudora mailbox (.toc and .mbx), PocoMail and Barca mailbox (.idx and .mbx), Opera mailbox (.mbs), Forte Agent mailbox (.idx), The Bat! mailbox (.msb and .tbb), Pegasus mailbox (.pmi, .pmm, and .cnm), PMMail message (.msg), FoxMail mailbox (.box), maildir folders (local copies), Mailbag Assistant mailbox (.mbg), MHT

Web Archive (.mht), and E-mail Examiner (.pmx). By default, X-Ways Forensics tries to extract from files matched by this filter expression: *.pst;*.ost;*.dbx;*.fdb;*.pfc;*.mbox;*.mbx;*.mbs;*.msb;*.tbb;*.pmm;*.cnm; *.mbg;*.mht;*.pmx;*.eml;*.msg. For more information please see the separate chapter about e-mail processing!

A forensic license allows to search for **JPEG and PNG pictures embedded in** documents such as MS Word, PDF, MS PowerPoint, MS Excel as well as in MP3 files, thumb*.db thumbnail buffers, Firefox cache container files and other files (e.g. *.doc;*.pdf;*.ppt;*.pps;*.xls;*.ole2; *.mp3;*.jpg;thumb*.db;_CACHE_*). Such pictures can be found by their file header signature if they are not stored in a complicated manner. They will be listed with generic names as "Embedded 1....jpg", "Embedded 2....png", etc. If the JPEG/PNG files are fragmented within the host file, they may not be fully viewable (appearing corrupted) or checkable for skin colors. If *.jpg is in the series of file masks you can find JPEG thumbnails incorporated in JPEG pictures. Only one JPEG picture will be searched per JPEG file.Thumbnails in certain 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.
This feature also searches and lists .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.

A forensic license allows to **extract** JPEG **pictures from video** files, in a user-defined interval (e.g. every 20 seconds). 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). Extracting pictures considerably reduces the amount of data, and looking at stills in the gallery is much faster 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. 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.

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.

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), .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.

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 hide and omit the file when trying again.

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*.

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.

X-Ways Forensics remembers for each file in the volume snapshot, which refinement operations have already been applied to it, so that it will not unnecessarily be touched again. You have the ability to reset selected files to the “still to be processed” status by pressing Ctrl+Del. This will also clear any computed skin color percentages, extracted metadata, hash values, hash matches, etc. etc. However, this function does not remove any child objects from the volume snapshot. That has to be done by the user before, by hiding and removing them.

3.6 Directory Browser

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. By default, the directory browser lists existing files and directories first, then deleted files and directories. 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. 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.

The directory browser can **sort** files and directories in ascending or descending order, and still reveals the previous sort criterion 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 disable the secondary sort criterion, 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.

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.

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.

Columns & 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. The following applies only if GREP syntax is *not* enabled: One filename mask can be a whole filename or a substring of a filename. If a substring, the missing part is substituted with an asterisk, 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 GREP syntax *is* enabled, then all the rules above do not apply. A search is run in the filenames for the specified GREP expression(s). For an explanation of GREP notation please see Search Options. The anchor $ does not work in this context.
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	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. (forensic license only)
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)
Status	The status of the preceding 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 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 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. (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 categorie 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 expression is interpreted as a substring that can match any part of the path, so no wildcards are needed or supported.
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)
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. On FAT, only the date is recorded. Filter available.
Record update*	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.
Deletion*	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 it 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.
Internal	Creation timestamp that can be extracted from the internally stored metadata in

creation	various file types (see context menu command). Internal timestamps are usually less volatile and more difficult to manipulate than file system level timestamps. They are useful for corrobation. Filter available. (forensic license only)
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. "Partial initialization" means that according to the NTFS file system 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.
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)
Hard links	The hard link count of the file or directory, i.e. how often it is referenced by a directory. (forensic license only)
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 e.g. to easily identify files that are obviously affected by ranges of bad sectors.
# files (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.
#ST (search 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 10 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, not limited to 10 search terms.
ID	The identifier assigned to the file or directory by the file system or by WinHex. Not necessarily unique.

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).
Dimensions	The size of a picture in thousand pixels (KP) or million pixels (MP, megapixels), as the result of width times height, rounded. KP values are displayed in gray, so that it's easier to recognize smaller pictures. The dimensions are computed simultaneously with skin color percentages, plus when viewing pictures (full-screen mode, preview mode, or in the gallery). Useful to easily distinguish between e.g. small browser cache garbage graphics and high-quality digital photos, with the associated filter. (forensic license only)
SC%	Skin color percentage. Available after refining the volume snapshot. Indicates the degree pictures are composed of skin tones. 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 and white pictures). (forensic license only)
Hash	The file's hash value, if computed.
Hash set	In the internal hash database, the name of the hash set that the file's hash value, if available, belongs to. Note that this is only the name of a single hash set even if the hash value is contained in multiple hash sets in the hash database. Filter available. (forensic license only)
Hash category	The category of the hash set that the file's hash value, if available, belongs to. Either "irrelevant", "notable", or blank. Filter available. (forensic license only)
Report table	The name(s) of the report table(s) that the file or directory has been assigned to. Filter available. (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 parentheses, 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.

*Please note that for FAT volumes, all timestamps are displayed unmodified, for all other volumes the time zone concept applies.

Most filters are available with a forensic license only.

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.

3.7 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. In the same dialog window you can also create new

report tables, rename or delete existing ones, and remove/override previous associations. You can associate the selected file or directory to a report table and/or at the same time the selected file's parent file (if any) and the file's or directory's child objects and any known duplicates in the same evidence object (which have been identified as duplicates based on hash values and marked as such in the Attr. column).

In order to output report tables to a report, use the Create Report command in the Case Data window.

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.

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.

3.8 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) plus the structure of Windows registry files, Windows Event Logs (.evt and .evtx), Windows shortcut liles (.lnk), Windows Prefetch files, $LogFiles, $UsnJrnl:$J, Windows Task Scheduler (.job), $EFS LUS, INFO2, Restore Point change.log.1, wtmp and utmp log-in records, MacOS X kcpassword, and AOL PFC 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 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, CorelDraw, 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 | External Programs. 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". You may also recursively explore all the keys and values in a hive and sort them in a chronological order via the virtual "Explore recursively" key.

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.

$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.

3.9 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.

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 EPOCHE (Unix) timestamp
%E EPOCHE8 (Unix) timestamp as QWORD.
%T Windows system time timestamp
%s ANSI-ASCII null-terminated
%S UNICODE string null-terminated
%b data not to be interpreted as characters (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
%b do not convert REG_BINARY to text
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".

3.10 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, MDI, PDF, RTF, WRI, AOL PFC, ASF, WMV, WMA, MOV, MP4, 3GP, M4V, M4A, JPEG, BMP, EXE/DLL (only on drive letters), THM, TIFF, GIF, PNG, GZ, ZIP, PF, IE cookies, DMP memory dumps, hiberfil.sys, PNF, SHD & SPL printer spool, WIM Vista image files, DocumentSummary alternate data streams, and tracking.log files. 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. The gallery does not go together very well with search hit lists.

Calendar (timeline view)

Gives a convenient overview of when the files/directories selected in the directory browser were created in a file system (red), last modified (blue), and last accessed (green), in the form of a calendar. Each day with a time stamp for at least one file or directory is filled in the calendar with the corresponding color. Weekends (Saturdays and Sundays) are specially marked. Hover the mouse over a day to find out which files exactly are represented and to see the corresponding times. If the list for a certain day is too lengthy to be displayed completely, you can still sort the directory browser in a suitable way and find out there.

Example: During which period of time were JPEG files created 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, then select all listed files, enable the calendar view, and watch out for red bars.

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).

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.

3.11 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 physically or logically. 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 Unicode (UTF-16LE) and in up to two 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.

It is possible to review the (incomplete) search hit list in the middle of an ongoing simultaneous search. Clicking the search hit list button will pause the search and allow to view the preliminary search hit list, until resuming the search if necessary. Useful e.g. when working 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.

3.12 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 contained in PDF (Adobe), WPD (Corel WordPerfect), CDR (Corel Draw), VSD (Visio), SWF (Shockwave Flash) and files of other formats supported by the viewer component can automatically be extracted/decoded/decompressed prior to search, to unformatted 8-bit ASCII or 16-bit Unicode plaintext, which can be reliably searched in addition to the actual file contents themselves. 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;*.eml;*.wpd;*.cdr;*.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.

- 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 hidden 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 included if the file slack option is enabled, as that option has a higher priority.

- 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 PST 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 e-mails and files will be searched, in the 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 such 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.

- The blind spot that logical searches have in old-fashioned computer forensics software products in the several thousand dollar price range (that they do not cover the transition from file slack to directly following free space) does not exist in X-Ways Forensics, as such areas on a partition can be addressed specifically.

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 hide and omit the file when trying again.

3.13 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. In that mode of operation is consists of three additional columns: physical/absolute offsets of the search hits, logical/relative offsets, 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. To get into that display mode, click the button with the binoculars and the 4 horizontal lines. It is only available for evidence objects.

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 file" option).

Search hits can be marked as notable (such that a 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. The search term list allows to create a quick overview of all hits marked as notable.

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.

3.14 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 used for conventional (non-index) searches in the case, plus those index search terms for which index search hits have been permanently saved.

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 file only” and then click Enter. This can be very useful if you are going to review all such files manually, as that ensures 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 option and click the Enter button again.

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.

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).

3.15 Indexing, Index Search

Available only with a forensic license, in the Search menu, when working with a case, for evidence objects. 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.

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 metadata output folder of the corresponding evidence object. The scope of the index, i.e. which files are to be indexed, can be fine-tuned. The default setting is that all existing files (including their slack, unless disabled in the directory browser options) plus the virtual files (which includes all free space) will be indexed. This avoids that certain parts of free space are indexed multiple times if they are referenced by several deleted files at the same time. 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 execption 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 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 selected computer media/images associated with a case in a single step.

You can index in Unicode and in up to two different code pages simultaneously. Please note that X-Ways Forensics cannot simultaneously index characters in the same multi-byte character code page if some characters utilize just 1 byte and others 2 bytes.

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
ë>e
й>и
...
```

(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.

Distributed indexing: Allows to accelerate index creation in time-critical cases. If *n* computers open the same case file (from a shared network drive) and participate in indexing the same evidence object(s), each computer can index approx. 1/*n* of the total data (may vary depending on the size of very large files within the volume snapshot). If all resulting index files (.xfi files) are created or eventually collected in the same metadata folder, they are treated exactly like an index created by just one computer. To ensure that no part of the volume snapshot is indexed twice or accidentally left out, all participants need to agree on the same index settings and get unique numbers assigned. E.g. if 9 computers are involved, each of the numbers 1...9 needs to be specified for indexing exactly once.

When using distributed indexing, X-Ways Forensics tries to detect differences in the index settings used by the various participants (options such as code pages, substring support, character pool etc.). If detected, at least one of the participants will be warned before indexing starts on that machine. Obviously, in a shared indexing effort the settings should be same everywhere.

When multiple examiners share the same image file, yet each work with their own case file because they examine different aspects of the same case, or when providing non-IT examiners with evidence file containers and pre-compiled search indexes, or when using the distributed

indexing feature to accelerate index creation, there is the option to have a common metadata subdirectory with the search index, which saves drive space, accelerates access because of synergetic file buffering in Windows, and facilitates handling of the search index files. Such a shared metadata directory for search index files (.xfi files) is used for both index creation and index search, however only if it is specifically created by the user, i.e. if it exists when needed. It is expected as a subdirectory of the directory where the image file is located, with the same base name as the image files, without extension, and the suffix " Metadata". E.g. if the name of the image is "Smith HD1.e01", then the expected name of the corresponding subdirectory is "Smith HD1 Metadata". If you prefer to store the index files on a different drive for performance reason, simply create the metadata directory as an NTFS reparse point that redirects to a different drive, but this and whether this feature is used at all is at the user's discretion.

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 hide and omit the file when trying again.

Search in Index: After indexing files, you may search the index for keywords very quickly. All files with the extension .xfi in the metadata subdirectory of the respective evidence object will be searched. Type in one or more search terms (1 per line) and start the search. Anything in excess of the maximum word length used for indexing is ignored. X-Ways Forensics does not distinguish between uppercase and lowercase letters except if a case-sensitive index was created. If listing search hits takes too long, e.g. because you entered a single character only or a very frequent short word, you may press Esc or close the progress indicator window to abort. Attention: If you select to search for your search terms also *within* words, but did not prepare the index for substring searches, then the result will be incomplete and slow. In the search hit list, physical offsets are not available.

If you wish to search the indexes of multiple or all evidence objects in a case at the same time, invoke Search | Search in Index from within the *case root* window and make sure these evidence objects are selected for the recursive listing and have been indexed.

3.16 Index Optimization

Optional step that can be run once an index has been created and that is executed automatically after indexing. You can safely abort the optimization at any time if you wish to continue using the program yourself (i.e. for an index search). During optimization, the various index*.xfi index component files will be consolidated/merged/unified to fewer uindex*.xfi files, finally to only a single .xfi file, which will be somewhat more efficient to search.. Also ensures that the Export Word List feature won't export duplicate words.

Two parameters control the optimization. These parameters can be changed even after optimization starts, and the changes take effect the next time a new optimization *process* is started, which usually happens every few minutes. The first parameter controls the amount of

memory used by a single process, which can be between 300 MB and 2 GB. This value should not exceed the amount of memory installed in your system. The more memory you allow an optimization process to use, the better the resulting optimization, because more .xfi files are compiled into single search tree. This translates to fewer word duplicates and better search performance, at the cost of more time for optimization. The second parameter controls the number of processes that optimize the index in parallel. This value should not exceed the number of CPU cores present in the system. Configuring the number of optimization processes is useful if you want to use the computer for other tasks while optimizing an index.

Question: What is the purpose of index optimization?

Answer: Index optimization eliminates duplicate words in the index and writes the index in an optimized file format. An optimized index requires less space on disk and can be searched faster.

Question: How much memory should I assign for index optimization?

Answer: Increasing the amount of memory available for index optimization improves the ability to eliminate duplicates from the index. On the downside, optimization takes longer if more memory is utilized. Testing index optimization with different settings allows you to gain experience which settings work best for your system. Index optimization can use more than 4GB if it runs on a 64-bit variant of Windows XP, Windows Vista, or Windows 7.

Question: Can indexing utilize more than one core?

Answer: Yes, indexing can use more than one core. This is particularly true of index optimization, which under optimial conditions will run n times faster on a system with n cores. Note that swapping can slow down optimization if not enough RAM is available for indexing.

Question: How can indexing help in cracking passwords?

Answer: It is possible to generate a word list from an index, which can be used with third-party tools for gaining access to encrypted data. In this case, we advise to optimize the index with as much RAM as possible, to reduce the number of word duplicates. Unlike for index searches, we advise to create a case-sensitive index for customized dictionary attacks on passwords (option "Match case").

3.17 Hash Database

Functionality only available with a forensic license. The 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. The 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).

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 does not check whether the matching hash sets all belong to the same category (which they should).

The Tools menu allows you to

- manage the active hash database: create a new (empty) one, view the list of hash sets, rename and delete hash sets, toggle the hash set category, and verify the integrity of the hash database (F8)
- import a single hash set text file (NSRL RDS 2.x, HashKeeper, and ILook text files as well as * are supported)
- 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
- delete the active hash database, e.g. to start a fresh one with new hash sets and/or a new hash type.

The Create Hash Set command in the directory browser's context menu allows you to create your own hash sets in the internal hash database. 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.

*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, one per line. Line break is 0x0D 0x0A.

3.18 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 for converting UTC to local filetimes. 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.

Export lists are output in local time.

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.

3.19 E-Mail Processing

Part of refining the volume snapshot.

E-mail messages are usually output as .eml files. Exceptions: E-mail messages extracted from OST and PST files (PST only when not using MAPI) may be occasionally extracted as plain text files or HTML files. Note that these two file types do not belong to the category e-mail, but you can easily and conveniently focus on all extracted e-mail messages from all e-mail archives if you explore recursively and use the Attribute filter instead of 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 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. Optionally, attachments can additionally be embedded directly in the listed .eml files as Base64 code (except for OST archives and PST if processed with the non-MAPI method). This allows to directly view pictures embedded in HTML e-mails and may be useful when viewing .eml files outside of X-Ways Forensics, but is not recommended because it takes more time, requires more drive space, and slows down indexing unnecessarily because of how Base64 coding works. Also it prevents attachments from becoming direct child objects of their respective parent e-mail messages in the volume snapshot. This option has no effect when dealing with AOL PFC, OST, and (if the MAPI method is used) PST files.

You can indicate a preference about how PST e-mail archives are processed. PST e-mail archives can be processed either through the MAPI interface or not. If the preferred method fails, the other method is attempted automatically. If the preferred method is MAPI, the non-MAPI method is still used to find traces of e-mail messages in unallocated space within the PST files. MSG files are always processed through MAPI. MAPI processing requires a fully functioning Extended MAPI system, as it comes with MS Outlook (32-bit version). Only Outlook 2003 and newer are capable of processing the Unicode variant of PST e-mail archives. Non-Unicode PST files can be processed with earlier versions of MS Outlook. Outlook 97 and older are not recommended and may not work correctly. If X-Ways Forensics always fails to extract PST e-mail archives through MAPI, go to the Windows Control Panel and check for an e-mail profile named like "pstloadtmp000" and delete it. It may happen that in artificially generated PST archives subject lines with Bates numbers in some e-mail messages do not retain these numbers when extracted.

The extraction method without MAPI is generally faster, has a good chance to work on PST archives where MAPI fails, has a chance to find traces of deleted e-mail, provides more data from Calendar/Contact/Notes/Tasks/Journal entries, fully supports non-English Unicode characters, supports overlong paths, and can process password-protected PST archives without the password! Even if you use MAPI for PST files, this alternative extraction method is used additionally to find and extract deleted contents. The same alternative extraction method is always used for OST files. The alternative extraction method 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.

3.20 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 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.

Evicence file containers can be interpreted, added to a case and conveniently examined like other image files with X-Ways Forensics 12.85 and later, and in particular also in X-Ways Investigator, the simplified version of X-Ways Forensics for investigators that are not computer forensic examiners, but spezialized 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.

When creating a new 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.

You may either optimize the container for better performance, when about to add a really great number of files, or prefer a slim container for fewer 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 active file container 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 choose whether to also copy child objects of selected directories and files.

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. If has an effect on the respective *parent* directory of selected items only if you enable an additional option (“Include parent item data/contents”). 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 the option “Include parent item data/contents” is enabled, even if the respective parent is a *file* that file's contents will be automatically copied (e.g. the e-mail message that the selected attachment belongs to, the zip archive that contains the selected file, or the document that the selected picture is embedded in).

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, it can be permanently suppressed.

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 or taken from the volume snapshot, if available.

Optionally, the preparer of an evidence file container can pass on his or her own report table associations or comments about included files with the container, via a subdirectory in the path where the container file is created. The recipient of the container will see those comment if he/she is not only provided with the container, but also with the optional metadata subdirectory of that container, when adding the container to the case. 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. If you select to pass on comments with the container when creating it, fill it with files that have comments, and then close the container, the metadata subdirectory is created automatically, with the same base name as the container, without extension, and the suffix “ Metadata”. E.g. if the name of the container is “Smith HD1.ctr”, then the name of the corresponding subdirectory is “Smith HD1 Metadata”.

When closing a container that is open in the background, the user is offered to compress, encrypt, and/or split it. This is useful if the container is complete and relatively huge, and e.g. should be

sent to someone else on CDs or DVDs.

3.21 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-hidden 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.

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. For other files, the mode of operation depends on the installed components: If X-Ways Trace is installed, and the file is either the Internet Explorer’s “index.dat” or Mozilla's/Firefox's “history.dat” or Opera's “dcache4.url” file, X-Ways Trace is invoked for these files. If the X-Ways Forensics separate viewer component is active, all other files are sent to that viewer. If it is not, the first installed external program will be called instead. 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.

Exceptions to all of the above are files beyond 2 GB in size and NTFS system files. These are always opened as data windows.

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").

External Programs

Allows sending 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.

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.
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 path from the currently explored directory). 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).
- 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 if you would not be able to access (e.g. view, copy or delete) such files anyway (because ordinary tools like the Windows Explorer do not allow that).
- 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) are re-applied to the recovered/copied files.
- Duplicate filenames will be changed to unique filenames by inserting incrementing numbers before the extension.
- 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. See 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 directories and files, also 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. Either a suffix character of your choice is appended (and by default that is a special Unicode character that is invisible in most 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).
- 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.
- 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.

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. 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).

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.

Extract Internal Metadata

Requires a forensic license.
1) 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. Additionally this command populates the

Sender and Recipients columns for original .eml files.
2) 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, 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.

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.

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*.

Hide/Unhide

You may hide selected items or hide all tagged or all untagged items. If actually filtered out, hidden files are excluded 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 hide all files in all other directories to ensure that. Refining the volume snapshot can be limited to files that are not hidden. Hidden 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 unhidden 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 Hide | "**Duplicates in directory browser based on hash**" to hide duplicate files from the currently listed part of a volume snapshot, based on hash values (if hash values were calculated). Only one out of two or more identical files will not be hidden. Do not apply this command more than once to the same files, or else *all* identical files might be hidden, 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 hidden. Identical attachments (child objects) will be marked as duplicates, but they will be hidden only indirectly if they are part of identical e-mail messages and

those are hidden, 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 hidden.
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.

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.

Position

The Position group of commands allows interactions with the currently selected file on a generally more technical level. It allows accessing the file's (or directory's) first cluster on the disk in the sectors view, accessing its related information like MFT record in NTFS or Inode in Ext2/Ext3/Ext4 and also sorting the files by their physical order on disk: "Sort by directory entry location" (FAT), "Sort by Inode Offset" (Ext2/Ext3/Ext4) or "Sort by MFT ID" (NTFS), respectively, allow to see files and folders in the order in which they physically appear in file system data structures (directory entries, the MFT, or Inode tables).

The Position 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.

Create Hash Set

Creates a hash set of the currently selected files and directories and their subdirectories directly within the internal hash database.

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 metadata directory, 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 select object.

Rename

Ability to rename virtual directories and virtual attached files in a volume snapshot.

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. 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.

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.

Save hit permanently

In a search hit list filled with index search hits, allows to permanently save selected hits under the search term used. By default, index search hits are not saved, as they can be listed again easily within a few seconds, due to the nature of index searches.

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.

4.2 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.

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. “Backups”

Restore Image: Select an image or backup file (.whx file) that you would like to restore. Image files will be first interpreted (which requires a specialist license) and then preset as the source in the dialog window “Clone Disk”. Raw images that are not split could also be restored without a specialist license, invoking the Clone Disk command directly.

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 rang 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: Lets you edit the size, the time stamp and attributes of a file (under Windows NT as well of a directory). Valid attributes are: A (archive), S (system), H (hidden), R (read-only). After entering new values in any area (size, time or attributes), simply press the **ENTER** key, so the modifications take effect.

Open Folder: 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.

4.3 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.
- **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.

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.

Convert: cf. Conversions

Modify Data: see below

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

4.4 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 occured 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.

4.5 Position 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. Sector and cluster numbers may optionally be entered in hexadecimal notation (with the 0x prefix). Please note that the data area on FAT drives starts with cluster #2.

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

4.6 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 characters set for the text column. Keyboard input is supported only for ANSI and IBM ASCII. You may also use **SHIFT+F7**. ANSI ASCII is the default character set. Unicode characters (little-endian) are always expected at even offsets.

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 skip to the next or to the previous byte value difference by clicking the extra 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.

4.7 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 files from different paths in a single step. For example, this allows to copy/recover 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 as recognized by WinHex. This can be useful to access surplus sectors at the end of the disk (in case they were not detected by WinHex), or to adjust the CHS coordinate system to your needs. Use this command on a logical drive to override the total number of clusters WinHex detects on that drive. This can prove useful when examining huge DVDs, which are detected as 2 GB media under Windows 9x.

Open RAM: See chapter “RAM 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.

4.8 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).

Wipe Securely: This command is used to erase the contents of one or more files irrevocably, such that they cannot be restored by WinHex or other special data recover software. Each selected file is overwritten according to the current settings, 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, which is 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. Note that you cannot apply this command to such a problematic directory itself, only to a parent directory.

4.9 Specialist Menu

Specialistand 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. Both the Create Disk Image command and the DOS cloning tool X-Ways Replica are able to image disks and produce canonically named file segments. Image segmentation is useful because the maximum file size supported FAT file systems is limited.
In some rare cases WinHex may be unable to correctly determine whether the first sector in an image is the sector that contains a master boot record or already a boot sector, and consequently interprets the image structure 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 original sector size. When the segments of a raw image are spread across two different drives, you may hold the Control key to be able to specify the other storage location. Should there be any problems with detecting the file system in a volume, you may hold both Ctrl *and* Shift while opening it to indicate the file system type you suppose in the volume yourself.
Mode 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 dynamic Virtual PC VHD images can be interpreted. Only allocated areas in such 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.

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.

4.10 Options Menu

General Options: see below

Viewer Programs: Here you may enable the separate viewer component and specify the path where it is located (by default: subdirectory ..\viewer). You may decide to use it for pictures, too (which is useful if you frequently print pictures, as the internal picture display cannot print). 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.

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. Please note that we cannot provide support for external programs.

You may also specify several 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.

Undo Options: see below

Security Options: see below

Data Interpreter Options: cf. Data Interpreter

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

4.11 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.

4.12 Help Menu

Contents: Displays the contents of the program help.

Setup: Lets you switch between the English, the German, the French, the Spanish, the Portuguese, and the Italian user interface.

Initialize: Use this command to restore the default settings of this 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 the WinHex homepage, the support forum, the Knowledge Base, or the newsletter subscription page in your browser.

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

4.13 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 Some Basic Concepts

5.1 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.

5.2 Entering Characters

In hex mode only hexadecimal characters are to be entered ('0'...'9', 'A'...'F'). In text mode you can enter all kinds of characters: letters, numbers, punctuation marks and special characters (e.g. '»', ']' and '^'). Please use the Windows program charmap.exe to find out key combinations for such characters (e.g. Alt-1-7-5 for '»'). The “WinHex” font even supports the Euro symbol (€).

5.3 Edit Modes

The info pane displays for each file/disk, in which mode it was opened. 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, only viewed. In other words, they are opened write-protected = read-only.

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 RAM editor.

Even in in-place edit mode the creation of a temporary file is unavoidable when altering the file size.

5.4 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.

5.5 Scripts

Most 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.

5.6 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.

5.7 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, and Windows dynamic disks as organized by the LDM (Logical Disk Manager). 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 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, 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.*

5.8 RAM Editor/Analysis

The RAM editor allows to examine the physical RAM/main memory and the logical memory of a process (i.e. a program that is being executed). 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 select one of the listed processes, you may access either the so-called primary memory or the entire memory of this process or one of the loaded modules. The primary memory is used by programs for nearly all purposes. Usually it also contains the main module of a process (the EXE file), the stack, and the heap. The "entire memory" contains the allocated page of the entire logical memory address space of a process, including the part of memory that is share among all processes.

Please note the following limitations:

- Access to physical RAM under Windows 2000/XP (32-bit) only, no more than 4 GB supported, 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 RAM 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) or a main memory dump file (and interpret that file exactly like you would a disk image), 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 64 bit.

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.

5.9 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.

6 Data Recovery

6.1 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".

6.2 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 Recovery by Type 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 Recovery by Type 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.

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 algorithm, and so are the original *filenames*. That is why the resulting files are 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: Exif JPEG files are optionally automatically named after the digital camera model that created them and their internal time stamp, if available. Many Windows Registry files are given their original names. Thumbs.db files are automatically named thumbs.db. The aforementioned prefix is not used in conjunction with original filenames.

The internal algorithm tries to determine the original size of JPEG, GIF, PNG, BMP, TIFF, Nikon NEF, Canon CR2 raw, PSD, CDR, AVI, WAV, MOV, MP4, 3GP, M4V, M4A, ASF, WMV, WMA, ZIP, RAR, 7Z, TAR, MS Word, MS Excel, MS PowerPoint, RTF, PDF, HTML, XML, XSD, DTD, PST, DBX, AOL PFC, Windows Registry, Prefetch, SPL, EVTX, and EML files by examining their data structure. The corresponding entries in the file type definition database must not be altered in order for the size and type detection to work for these file types. For files about whose original size the algorithm has no idea, the files are recovered at the exact default file size specified by the user in KB. Be generous when specifying this size because whereas files recovered "too large" can still be opened by their associated applications, prematurely truncated files however often can't be as they are incomplete. The attempt to detect the original size of files of certain types is limited by a maximum size, which is a user-supplied multiple of the default file size.

Technically it is possible to select as many file types for simultaneous recovery as you like. However, if you e.g. recover MS Office and AVI files at the same time and the MS Office files you expect are around a few KB and the AVI files around a 1 GB in size, using a single global default file size would not be a good idea. That's why optionally you can define individual default sizes for each file type in the file type definition database, which if exist optionally have priority over the global default file size specified in the dialog window.

File headers can be searched only at *cluster* boundaries, as the beginning of a cluster is the only place where a file can start in a cluster-based file system. However, you may also select to search for sector-aligned file headers. This is a good compromise, the default setting, the most optimized algorithm, and useful to find files from a previously existing volume 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 if cluster boundaries are selected. There is yet another possibility, a thorough byte-level search. This is necessary when recovering files from backup files or tapes, or JPEG files from within MS Word documents, where they are not aligned at cluster or sector boundaries. 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.

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.

When searching for MPEG file signatures at sector boundaries, the internal algorithm ensure that no overlapping MPEG fragments and no MPEG fragments in the middle of known MPEG files will be output/listed. This is useful because the MPEG signature occurs throughout MPEG files, not just at the start.

6.3 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.

5th column: Footer

Optional. A signature (constant 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. Still, 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.

6th column: Default in KB

Optional. A file type specific default maximum file size in KB that can override the global maximum file size specified in the File Recovery by Type dialog window. Useful because e.g. an MPEG video could be more around 1 GB in size, where a Windows icon file (.ico) could be around 1 KB in size.

6.4 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.

7 Options

7.1 General Options

1st column:

- 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).

- 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.

- 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.

- The option **Allow multiple program instances** lets you execute WinHex more than once at a time. If it is not enabled, WinHex puts the main window of the running instance into the foreground instead of creating a new program instance.

- **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.

- By default, **edit windows** are not **opened** in a **maximized** state.

- On a right click, **WinHex** can bring up a special **context menu**, the regular edit menu, or define the end of the current block. If this option is disabled, you can still bring up the context menu if you hold the Shift key while right-clicking.

- 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.

- 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. Only under Windows XP surplus sectors are included automatically, which renders this option obsolete.

- The **alternative access method 1** for physical hard disks under Windows 2000/XP 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, under Windows 2000/XP. 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**. You may specify just a period (.) as a placeholder to use the directory from where WinHex/X-Ways Forensics is executed. This also also possible for the next three 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**.

  X-Ways Investigator GUI/Reduced user interface: Available when operated with a forensic license. Activates the considerably reduced user interface of X-Ways Investigator, 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.

- 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.

- 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, 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.

- 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.

- When gallery view is enabled, WinHex can optionally continue **loading thumbnails in the background** when the current view is full. (option currently not available)

- 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.

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.

- **Offsets** can be presented and prompted for in a decimal or **hexadecimal** notation. This setting is valid for the entire program.

- When using the RAM editor it may be reasonable to have WinHex display **virtual addresses** instead of zero-based offsets. This is always done in hexadecimal notation. The dialog window of the Goto Offset command will also prompt for virtual 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.

- Specify how many **lines to scroll** when **rolling** the mouse **wheel** (if available).

- **NTFS: MFT auto coloring:** Highlights the various elements in FILE records of the NTFS file system, when the blinking cursor is located within such a record, to facilitate navigation and understanding. Requires a specialist or forensic license.

- 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 ANSI ASCII mode. The WinHex font implements the full Windows character set (even characters such as the ™ and € symbols and diverse quotation marks).

- Last not least, you may select one of several different **dialog window** and **button styles**.

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 years in dates with 2 digits only.

- 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.

Factory settings of *all* options can be restored using the Initialize command of the Help menu.

7.2 Directory Browser Options

- **Grouping files and directories** in the directory browser is optional.

- **Grouping existing and deleted items** in the directory browser is optional. There are two possibilities how to enable this feature, either potentially recoverable deleted files (marked with a question mark) and known unrecoverable files (marked with an X) are internally grouped as well or not.

- Optionally, the e-mail **header** of **.eml** files can be **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.

- **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**.

- 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.

- 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, to save time. That means the program will forget the last sort criteria in use last time.

- **Dynamic e-mail 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.

- **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.

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. Similary (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, duplicates within the same volume will be marked as already viewed at the same time, but this is potentially slow when used in conjunction with the gallery.
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.

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.

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.

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. 3rd checkbox state: Only items whose first cluster is known to be unavailable.
- 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.
- Hidden files and directories (marked as hidden in the volume snapshot).
- Files and directories that are *not* marked as hidden.

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.

Below the filter options in the lower left corner you will find a button in this dialog box that allows to unhide all files and directories in the volume snapshot of the evidence object in the active data window. To selectively unhide files, make sure they are not filtered out. Then you can unhide them with a context menu command after selecting them.
There is another button that allows to totally remove hidden 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. Available only for volume snapshots created by v14.2 and later. Useful also if you would like X-Ways Forensics to find certain files once again via a file header signature search, but 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. Hidden items that have non-hidden 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.

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.

7.3 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.

- Various volume snapshot refinement options will produce child objects for files, e.g. extracted pictures. If you prefer not to have **child objects** for **files**, but for directories only, auxiliary virtual directories can be created that act as the parent of the child objects, as in earlier versions. This is useful if you create containers that should be examined with earlier versions

of X-Ways Forensics or X-Ways Investigator. If you *allow* files with child objects (default), the Recover/ Copy command prevents name conflics in the output folder automatically.

- **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 exist. 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.

- **Keep more data** of the volume snapshot **in memory**, e.g. for much quicker sorting by timestamps.

- **NTFS:** Optionally **search for FILE records everywhere** as part of the particularly thorough file system data structure search.

- You can indicate whether you are interested in getting files listed whose clusters (and therefore data) are totally unknown, with only metadata (e.g. just filename), in NTFS, Ext*, and Reiser*.

- You can indicate whether you are interested in **earlier names and locations** of renamed/moved files and directories in NTFS and whether you are interested in getting **files** listed for which only filename, size, timestamps and attributes (but **no** data/**clusters**) are known. Affects the thorough file system data structure search (specialist license or higher). If the checkbox for earlier names/paths is half checked, then you will be informed of earlier names/paths of renamed/moved files via comments and don't get additional files in the volume snapshot for each earlier name/path.

- **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.

- 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

7.4 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.

7.5 Security & Safety Options

Use the option **Check for virtual memory** alteration to make sure the RAM 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**.

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.

All notices and warnings output to the **Messages** window can optionally be automatically saved in a text file “**messages.txt**” in the installation directory. If at that time a case is active, the

notice/warning will be written to the messages.txt file in the log subdirectory of that case instead.

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.

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 will at least 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.

7.6 Search Options

Case sensitive: By default a search is case-sensitive, so 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 can be fully case insensitive only with the Simultaneous Search, with the Find Text command only for letters from the Latin/English alphabet and German umlauts.

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 two different 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.

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 and Logical Search only. Regular expressions are a powerful search tool. A single regular expression may match many different words. 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. Only English language letters (a-z, A-Z) and German and French letters are considered word characters. The start and end of a file also count as word boundaries. With the new search algorithm, \b is only supported at the start and/or at the end of the search term. With the new search algorithm, anchors (\b, ^, $) only work only when searching in evidence objects of a case. With the old search algorithm \b did not work in Unicode.

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). E.g. \x0D\x0A is the Windows line break.
?	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.

Examples:

[a-zA-Z0-9_\-\.]{1,20}@[a-zA-Z0-9\-\.]{2,20}\.[a-zA-Z]{2,7}	finds e-mail addresses
[a-zA-Z]+://[a-zA-Z0-9/_\?$&=\-\.]+	finds Internet addresses with http://, https://, ftp://

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*).

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. (this paragraph quoted from wikipedia.org)

7.7 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.

8 Miscellaneous

8.1 Block

You can mark a part of an open file 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.

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.

8.2 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.

8.3 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*

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.

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.

8.4 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.

8.5 Disk Cloning

The command “Clone Disk” is part of the Tools menu. This function copies a defined number of sectors from a source to a destination disk, or alternatively from a raw disk image file or to a raw disk image file. Both disks must have the same sector size. In order to effectively *duplicate* a medium (i.e. in order to copy all sectors), enable 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 source disk you can also select any image (e.g. .e01 evidence file as supported by X-Ways Forensics) that is interpreted/treated internally as a disk, or a partition opened from within a physical disk in the background.

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 (25 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 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, and it is even possible to automatically fill an existing incomplete conventional ("forward") image additionally backwards to get an image that is as complete as possible, with only a small zeroed gap in the middle that represents the unreadable damaged spot on the source hard disk. Be sure to create reverse images on NTFS volumes, not FAT32.

For disk imaging in general it is recommended to use the File | Create Disk Image functionality 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 when 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 detailled control over which sectors are copied from where to where in which order.

8.6 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 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.

When creating raw image files or .e01 evidence files of volumes/partitions with compression, there is an option to store free clusters as zero-value bytes. That is useful if you create the image for data backup and not for forensic purposes, in conjunction with compression, to save drive space, or for forensic purposes if you are supposed to examine existing files only anyway. (specialist or forensic license only) Note that in case of file system inconsistencies clusters could be erroneously regarded as free. You have to specifically confirm this option as it will create images that in the traditional sense are not forensically sound.

Forensic licenses: 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.

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.

The encryption algorithm optionally used in .e01 evidence files is 256-bit AES/Rijndael, in counter (CTR) mode. This allows for random read access within evidence files. 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. The 128-bit counter is randomized and incremented per encryption block. The block size of AES is 128 bits. An additional SHA-256 is stored in the header as well 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*.

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.

8.7 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.

8.8 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.

8.9 Reconstructing RAID Systems

WinHex and X-Ways Forensics can internally destripe RAID level 0 and level 5 systems as well as JBOD consisting of up to 16 components (physical hard disks or images). That way it is not necessary to use scripts that unstripe and export RAID systems to a new image, saving you time and drive space. 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, data is not only striped across all component disks in a rotating pattern, but also interspersed with parity blocks for redundancy. RAID level 5 is implemented in different ways by different RAID controller manufacturers in that they employ different stripe/parity patterns. The supported patterns are the following:

Backward Parity a.k.a. Left Asynchronous (Adaptec)

```
Component 1:  1  3  P
Component 2:  2  P  5
Component 3:  P  4  6
```

Backward Dynamic Parity a.k.a. 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
```

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
```

Forward Parity (a.k.a. Right Asynchronous)

```
Component 1:  P  3  5
Component 2:  1  P  6
Component 3:  2  4  P
```

Forward Dynamic Parity (a.k.a. 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
```

The parity start component can be defined differently if necessary. 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"!

8.10 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 because 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/.

8.11 Data Interpreter

The Data Interpreter is a small window that offers possible translations for the data at the current cursor position. Contrary to popular believe 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 bits of a byte), four floating-point data types, assembler opcodes (Intel®), and date types. Dates are always represented in their original state exactly as stored. They are never converted to any local time.

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. This plus the digit grouping can also be selected in the Data Interpreter Options dialog.

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.

8.12 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).

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
```

These modifiers only affect the immediately following variable. They are redundant if they appear in the header already.

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 **1100** = **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.

Whereever 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 opional. 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
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